xeCJK 宏包

ctex.org

2013/12/05 v3.2.8*

目录

1 简介 1

2 基本用法 2

3 用户手册 2
- 3.1 宏包选项 2
- 3.2 字体设置与选择 5
  - 3.2.1 XeTeX 的字体名查找 . . . 7
- 3.3 CJK 分区字体设置 8
- 3.4 设置 CJK 字符范围 9
- 3.5 标点符号的处理 9
  - 3.5.1 设置特定标点符号的宽度和间距 9
  - 3.5.2 定义标点符号处理格式 . . 10
- 3.6 其它 12

4 已知问题和兼容性 12

5 xeCJK 代码实现 13
- 5.1 运行环境检查 13
- 5.2 内部工具 14
- 5.3 功能开关 19
- 5.4 字符类别设定 19
- 5.5 字符类别处理 24
- 5.6 字符输出规则 27
- 5.7 全角右标点后的断行 38
- 5.8 段末孤字处理 40
- 5.9 增加 CJK 子分区 43
- 5.10 标点处理 45
- 5.11 后备字体 56
- 5.12 CJK 字体族声明方式 57
- 5.13 字体切换 63
- 5.14 数学字体设置 69
- 5.15 抄录环境中的间距调整 70
- 5.16 xeCJK 其它选项 75
- 5.17 xeCJK 初始化设置 76
- 5.18 兼容性修补 78
- 5.19 xeCJKfntef 84
- 5.20 xeCJK-listings 92
- 5.21 xunicode-addon 97
- 5.22 xeCJK.cfg 107

版本历史 108

代码索引 109

1 简介

xeCJK 是一个 XeLaTeX 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 XeTeX 的一些最新特性，需要 XeTeX `0.9995.0` [2009/06/29] 以后的版本。xeCJK 依赖 everypage 和 LaTeX3 项目的宏包套件 l3kernel 和 l3packages 。xeCJK 还需要通过 fontspec 宏包来调用系统字体。将在 3.1 节介绍的功能选项 `indentfirst` 选项需要 indentfirst 宏包的支持。xeCJK 会自动根据需要载入这些宏包。

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

*`ctex-kit` rev533.

[1] `leoliu.pku@gmail.com`

[2] `sobenlee@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 ⟨*key*⟩=⟨*var*⟩ 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。

`\xeCJKsetup` `\xeCJKsetup {⟨key₁⟩=⟨var₁⟩, ⟨key₂⟩=⟨var₂⟩, ...}`

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 ⋆ 号，这表示这个选项或命令只能在导言区中使用，而 ☆ 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。

`LocalConfig` ⋆ `LocalConfig = {⟨true|false|name⟩}`

New: 2012-11-22

是否使用本地配置文件 `xeCJK-⟨name⟩.cfg`。⟨*name*⟩ 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-⟨name⟩.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-⟨name⟩.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。

indentfirst ★

Updated: 2012-11-22

`indentfirst = ⟨true|false⟩`

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。

请注意，xeCJK 宏包中只有上述 `LocalConfig`、和 `indentfirst` 这两个选项需要在调用 xeCJK 时设置，而不能通过 `\xeCJKsetup` 来设置。

xeCJKactive

`xeCJKactive = ⟨true|false⟩`

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 = ⟨true|false⟩`

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 = ⟨true|false⟩`

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。url 宏包将一个 URL 放在一个特殊的数学环境中排版，所以如果在 `\path` 等命令的路径参数中含有汉字，则需要启用这个选项，路径中的汉字才能显示。

CJKglue

`CJKglue = {\hskip 0pt plus 0.08\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 = {⟨glue⟩}`

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 = {⟨true|false|glue⟩}`

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

Updated: 2013-06-26

`CheckSingle = ⟨true|false⟩`

是否避免单个 CJK 文字单独占一个段落的最后一行。需要说明的是，这个选项只有在段末的最后一个字是 CJK 文字或者标点符号，并且倒数第二和第三个字都是文字才能正确处理处理孤字的问题。如果这倒数三个字有作为控制序列的参数的情况，那么一般来说也不能正确处理。

PlainEquation

New: 2012-12-06

`PlainEquation = ⟨true|false⟩`

如果使用了 `$$...$$` 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 `\[...\]` 的形式来输入行间数学公式。

NewLineCS
NewLineCS+
NewLineCS-

New: 2012-12-04

`NewLineCS = { \par \[ }`

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-

New: 2012-12-04

`EnvCS = { \begin \end }`

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+
InlineEnv-

Updated: 2012-12-06

`InlineEnv = {⟨env₁⟩, ⟨env₂⟩, ⟨env₃⟩, ...}`

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。

AutoFallBack

`AutoFallBack = ⟨true|false⟩`

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 = {⟨true|false|数字⟩}`

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 = {⟨true|false|数字⟩}`

全局设定当没有声明对应的斜体时，是否使用*伪斜体*；当输入的是数字时，将使用伪斜体，并将使用输入的数字作为伪斜体的默认倾斜程度。

EmboldenFactor ☆

`EmboldenFactor = {⟨数字|4⟩}`

设置伪粗体的默认粗细程度。

SlantFactor ☆

`SlantFactor = {⟨数字|0.167⟩}`

设置伪斜体的粗细程度，范围是 $-0.999 \sim 0.999$。

PunctStyle

Updated: 2012-11-10

`PunctStyle = {⟨quanjiao|banjiao|kaiming|hangmobanjiao|CCT|plain|...⟩}`

设置标点处理格式。xeCJK 中预先定义好的格式为

quanjiao	全角式：所有标点占一个汉字宽度，相邻两个标点占 1.5 汉字宽度；
banjiao	半角式：所有标点占半个汉字宽度；
kaiming	开明式：句末点号用全角，其他半角；
hangmobanjiao	行末半角式：所有标点占一个汉字宽度，行首行末对齐；
CCT	CCT 格式：所有标点符号的宽度略小于一个汉字宽度；
plain	原样(不调整标点间距)。

可以使用 3.5.2 中介绍的 `\xeCJKDeclarePunctStyle` 定义新的标点格式。

KaiMingPunct ★
KaiMingPunct+ ★
KaiMingPunct- ★

`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。

LongPunct ★
LongPunct+ ★
LongPunct- ★

`LongPunct = {⟨ —— ⋯⋯⟩}`

设置长标点，例如破折号“——”与省略号“……”，允许在长标点前后断行，但是禁止在它们之间断行。

MiddlePunct ★
MiddlePunct+ ★
MiddlePunct- ★

`MiddlePunct = {⟨ —— · · · ⟩}`

设置居中显示的标点，例如间隔号“·”。对于在 CJK 文字之间的居中标点，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。

PunctWidth ★

`PunctWidth = {⟨length⟩}`

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 `\xeCJKsetwidth`。

PunctBoundWidth ★

New: 2013-08-22

PunctBoundWidth = {⟨*length*⟩}

与以上选项类似，但设置的是标点符号出现在行首/尾时的宽度。

AllowBreakBetweenPuncts

AllowBreakBetweenPuncts = ⟨*true*|false⟩

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

New: 2012-12-02

CheckFullRight = ⟨*true*|false⟩

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

Verb

Updated: 2013-11-16

Verb = ⟨*true*|*false*|env|*env+*⟩

true 表示在 \verb 命令或 verbatim 环境里不自动调整中英文之间的间距。env 选项在 verbatim 环境里自动计算中西文间距和中文之间的间距，以便于保持代码的对齐；env 选项不调整 \verb 里的间距，env+ 选项还将正文里设置的间距应用到 \verb 里。这个选项对使用到 \verbatim@font 命令的情形均有效，更一般的情况可以使用 3.6 节介绍的 \xeCJKVerbAddon。false 表示不作任何处理。以上选项的值除 false 外，都禁止在汉字之间和汉字与西文之间自动换行。

3.2 字体设置与选择

\setCJKmainfont ★

\setCJKmainfont [⟨*font features*⟩] {⟨*font name*⟩}

设置正文罗马族的 CJK 字体，影响 \rmfamily 和 \textrm 的字体。后面两个参数继承自 fontspec 宏包，⟨*font features*⟩ 表示字体属性选项，⟨*font name*⟩ 是字体名。字体名可以是字体族名，也可以是字体的文件名，查找字体名见 3.2.1 节；可用的字体属性选项参见 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 = {⟨*true*|*false*|数字⟩}
AutoFakeSlant = {⟨*true*|*false*|数字⟩}

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

Mapping

New: 2013-06-07

Mapping = {⟨*fullwidth-stop*|*full-stop*|*han-trad*|*han-simp*|...⟩}

xeCJK 提供了以上四个 TECKit 映射文件，可以在设置字体的时候通过 Mapping 选项来使用它们。其中 fullwidth-stop 用于将正常句号“。”转换成全角实心句号“．”，full-stop 的作用相反。han-trad 用于将简体中文转换成繁体中文，han-simp 的作用相反。需要注意的是，简繁互换都是简单机械的字字对译，不能做到完全准确，使用时要小心。例如简体的“发挥”和“头发”被转换成繁体的“發揮”和“頭發”，显然后者应作“頭髮”。也可以根据实际需要，制作新的映射文件，请参考 TECKit 的文档。

\setCJKsansfont ★

\setCJKsansfont [⟨*font features*⟩] {⟨*font name*⟩}

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

\setCJKmonofont ★

\setCJKmonofont [⟨*font features*⟩] {⟨*font name*⟩}

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

\setCJKfamilyfont ★

\setCJKfamilyfont {⟨*family*⟩} [⟨*font features*⟩] {⟨*font name*⟩}

声明新的 CJK 字体族 ⟨*family*⟩ 并指定字体。

\CJKfamily

Updated: 2012-10-27

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

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

\newCJKfontfamily ⋆

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

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

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

等价于

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

\CJKfontspec

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

在文档中随机定义新的 CJK 字体族，并马上使用它。

\defaultCJKfontfeatures ☆

\defaultCJKfontfeatures {⟨*font features*⟩}

全局设置 CJK 字体族的默认选项。例如，使用

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

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

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

\addCJKfontfeatures

Updated: 2013-06-30

\addCJKfontfeatures {⟨*font features*⟩}
\addCJKfontfeatures * {⟨*font features*⟩}
\addCJKfontfeatures [⟨*block*$_1$, *block*$_2$, ...⟩] {⟨*font features*⟩}
\addCJKfontfeatures * [⟨*block*$_1$, *block*$_2$, ...⟩] {⟨*font features*⟩}

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

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

可以将文档中当前使用的 CJK 主分区字体放大为 1.1。

\CJKrmdefault

保存 \textrm 和 \rmfamily 所使用的 CJK 字体族，默认值是 rm。类似西文字体的 \rmdefault。

\CJKsfdefault

保存 \textsf 和 \sffamily 所使用的 CJK 字体族，默认值是 sf。类似西文字体的 \sfdefault。

\CJKttdefault

保存 \texttt 和 \ttfamily 所使用的 CJK 字体族，默认值是 tt。类似西文字体的 \ttdefault。

\CJKfamilydefault

Updated: 2013-01-01

保存 \textnormal 和 \normalfont 所使用的 CJK 字体族。类似西文字体的 \familydefault。初始值是 \CJKrmdefault。如果没有在导言区中修改它，xeCJK 会在导言区结束的时候根据西文字体的情况自动更新 \CJKfamilydefault。因此，在导言区里使用

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

就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。

\setCJKmathfont ★ \setCJKmathfont [⟨font features⟩] {⟨font name⟩}

设置数学公式中的 CJK 字体族。如果使用了 CJKmath 选项，但是没有使用 \setCJKmathfont 设置数学公式中的 CJK 字体，那么将使用 \CJKfamilydefault 作为数学公式中的 CJK 字体。

\setCJKfallbackfamilyfont ★ \setCJKfallbackfamilyfont {⟨family⟩} [⟨font features⟩] {⟨font name⟩}

设置 CJK 字体族 ⟨family⟩ 的备用字体。例如，使用

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

可以将 SimSun-ExtB 作为 SimSun 的备用字体。

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

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

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

如果 FallBack 的值为空，将设置的是备用字体。例如，

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

等价于

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

\setCJKfallbackfamilyfont ★

Updated: 2013-06-30

```
\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。若当前字体族缺字，并没有备用字体，则尝试使用 \CJKfamilydefault 的备用字体。

3.2.1 XeTeX 的字体名查找

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

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

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

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

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

```
Times New Roman:style=cursiva,kurzíva,kursiv,Πλάγια,Italic,
  Kursivoitu,Italique,Dőlt,Corsivo,Cursief,kursywa,Itálico,Курсив,
  İtalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta cursiva,tučné kurzíva,fed kursiv,
  Fett Kursiv,Έντονα Πλάγια,Bold Italic,Negrita Cursiva,
  Lihavoitu Kursivoi,Gras Italique,Félkövér dőlt,Grassetto Corsivo,
  Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrito Itálico,
```

```
  Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalın İtalik,
  Krepko poševno,nghiêng đậm,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Έντονα,Bold,Negrita,
  Lihavoitu,Gras,Félkövér,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
  Полужирный,Fet,Kalın,Krepko,đậm,Lodia
Times New Roman:style=Normal,obyčejné,Standard,Κανονικά,Regular,
  Normaali,Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,
  thường,Arrunta
宋体,SimSun:style=Regular
黑体,SimHei:style=Normal,obyčejné,Standard,Κανονικά,Regular,Normaali,
  Normál,Normale,Standaard,Normalny,Обычный,Normálne,Navadno,Arrunta
```

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

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

来设置字体。

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

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

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

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

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

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

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

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

3.3 CJK 分区字体设置

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

\xeCJKDeclareSubCJKBlock ★

\xeCJKDeclareSubCJKBlock {⟨*block*⟩} {⟨*block range*⟩}
\xeCJKDeclareSubCJKBlock* {⟨*block*⟩} {⟨*block range*⟩}

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

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

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

[3] 由于汉字编码原因，Windows 下总需要把字体列表输出的文件中防止乱码。

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

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

例如，可以使用

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

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 {⟨*block*$_1$, *block*$_2$, ...⟩}
\xeCJKCancelSubCJKBlock* {⟨*block*$_1$, *block*$_2$, ...⟩}

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

\xeCJKRestoreSubCJKBlock

\xeCJKRestoreSubCJKBlock {⟨*block*$_1$, *block*$_2$, ...⟩}
\xeCJKRestoreSubCJKBlock* {⟨*block*$_1$, *block*$_2$, ...⟩}

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

3.4 设置 CJK 字符范围

\xeCJKDeclareCharClass ⋆

\xeCJKDeclareCharClass {⟨*class*⟩} {⟨*class range*⟩}
\xeCJKDeclareCharClass* {⟨*class*⟩} {⟨*class range*⟩}

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

\xeCJKResetCharClass ⋆

用于恢复 xeCJK 对各个字符类别的初始化设置。

\xeCJKResetPunctClass ⋆

用于重置标点符号所属的字符类。

\normalspacedchars

\normalspacedchars {⟨*char list*⟩}

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

3.5 标点符号的处理

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

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 plain 以外的所有标点处理格式。

\xeCJKsetwidth ⋆

Updated: 2013-08-22

\xeCJKsetwidth {⟨标点列表⟩} {⟨*length*⟩}
\xeCJKsetwidth* {⟨标点列表⟩} {⟨*length*⟩}

⟨标点列表⟩ 可以是单个标点，也可以是多个标点。例如，

```
\xeCJKsetwidth{。？}{0.7em}
```

将设置句号和问号所占的宽度为 0.7 em。带星号的命令，设置标点符号出现在行首/尾时的宽度。

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

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

```
\xeCJKsetkern{：}{“}{0.3em}
```

将设置冒号与左双引号之间的空白宽度为 0.3 em。

3.5.2 定义标点符号处理格式

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

Updated: 2013-08-22

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

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

Updated: 2013-08-22

修改已有的标点符号处理格式。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

下面选项设置标点符号出现在行首或者行尾时的宽度或比例。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

就定义了一个名为 mine 的标点处理格式。可以在通过

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

在文档中使用这个格式。它的意义是:使用标点符号的实际左右边界中的最小值作为其左/右空白的宽度,对于句末标点和居中标点,再加上实际边界空白的一半;当标点出现在行首或行尾时宽度为零,允许悬挂出页面边界;使用相邻两个标点的实际边界中的较小值作为它们之间的空白宽度,并且最小的空白宽度是 0.1 em。再例如,使用

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

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

3.6 其它

\xeCJKVerbAddon
\xeCJKOffVerbAddon

Updated: 2013-11-16

调整文字间距以便于让 CJK 字符占的宽度等于西文等宽字体中两个空格的宽度。如果这两个空格的宽度小于当前 CJK 正常文字的宽度，将对 CJK 字体进行适当地缩小。这有利于等宽字体的代码对齐等情形。需要注意的是，\xeCJKVerbAddon 对 xeCJK 的内部进行了比较大的修改，使用它之后，将禁止在 CJK 字符类之间自动换行，这与西文在抄录环境中的情况是一致的。所以不应该单独使用，应该放在分组里限制其作用域，否则是无效的。当然它可以和其它关于代码抄录的宏包配合使用。例如，可以使用于 fancyvrb 宏包的 formatcom 选项。此时设置的西文字体应该确实是等宽的以保证对齐。若西文等宽字体发生变动（包括字体大小），则需要在其后面使用 \xeCJKVerbAddon，重新计算间距的宽度。\xeCJKOffVerbAddon 用于在使用 \xeCJKVerbAddon 的环境中局部取消它的作用。由于 listings 宏包有自己的代码对齐机制，所以 \xeCJKVerbAddon 在由 listings 定义的代码环境中无效。

\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。

\xeCJKShipoutHook

New: 2013-11-09

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

4 已知问题和兼容性

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

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

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

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

新版本(3.x)的 xeCJK 完全使用 LATEX3 的语法来编写。LATEX3 放弃了 \outer 宏的概念，因此相关工具在遇到 \outer 宏时可能会存在问题。按照目前 xeCJK 的实现方式，在 CJK 文字后面遇到 \outer 宏时会出现类似

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

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

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

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

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

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

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

5 xeCJK 代码实现

```
⟨*package⟩
⟨@@=xeCJK⟩
```

5.1 运行环境检查

xeCJK 必须使用 XeTeX 引擎的支持。

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

xeCJK_if_package_loaded_p:n
xeCJK_if_package_loaded:nTF

判断宏包是否被引入,可用于文档正文中。

```
\prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
  {
    \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
      { \prg_return_true: } { \prg_return_false: }
  }
\tl_const:Nx \c__xeCJK_package_ext_tl { \@pkgextension }
```

(*End definition for* \xeCJK_if_package_loaded:n.)

下面这些 CJK 系列宏包不应该被使用。

```
\msg_new:nnn { xeCJK } { incompatible-package }
  {
    The~`#1'~package~and~xeCJK~are~incompatible.\\\\
    Please~do~not~use~it.
  }
\msg_new:nnn { xeCJK } { after-package }
  {
    The~`#1'~package~and~xeCJK~are~incompatible.\\\\
    Please~load~it~after~xeCJK.
  }
\clist_map_inline:nn { CJKfntef , CJKnumb }
  {
    \xeCJK_if_package_loaded:nT {#1}
      { \msg_error:nnn { xeCJK } { after-package } {#1} }
  }
\clist_map_inline:nn { CJKulem , CJKvert , CJKpunct , CJKutf8 , CJK }
  {
    \xeCJK_if_package_loaded:nTF {#1}
      { \msg_error:nnn { xeCJK } { incompatible-package } {#1} }
      { \tl_const:cn { ver@ #1 . \c__xeCJK_package_ext_tl } { 9999/99/99 } }
  }
```

应该使用较新版本的 expl3 宏包。

```
\msg_new:nnn { xeCJK } { l3-too-old }
  {
    Support~package~`#1'~too~old. \\\\
    Please~update~an~up~to~date~version~of~the~bundles\\\\
    `l3kernel'~and~`l3packages'\\\\
    using~your~TeX~package~manager~or~from~CTAN.\\
    \str_if_eq:nnT {#1} { expl3 } { Loading~xeCJK~will~abort! }
  }
\@ifpackagelater { expl3 } { 2013/07/20 } { }
  { \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。

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

5.2 内部工具

分配临时变量。

```
\tl_new:N \l__xeCJK_tmp_tl
\int_new:N \l__xeCJK_tmp_int
\box_new:N \l__xeCJK_tmp_box
\dim_new:N \l__xeCJK_tmp_dim
\bool_new:N \l__xeCJK_tmp_bool
\skip_new:N \l__xeCJK_tmp_skip
\clist_new:N \l__xeCJK_tmp_clist
```

__xeCJK_msg_new:nn
__xeCJK_error:n
__xeCJK_error:nx
__xeCJK_warning:nx
__xeCJK_info:nxx

各种信息函数的缩略形式。

```
\cs_new_protected_nopar:Npn \__xeCJK_msg_new:nn   { \msg_new:nnn       { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_msg_new:nnn  { \msg_new:nnnn      { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_error:n      { \msg_error:nn      { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_error:nx     { \msg_error:nnx     { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_warning:n    { \msg_warning:nn    { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_warning:nx   { \msg_warning:nnx   { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_warning:nxx  { \msg_warning:nnxx  { xeCJK } }
\cs_new_protected_nopar:Npn \__xeCJK_warning:nxxx { \msg_warning:nnxxx { xeCJK } }
\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:

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

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

__xeCJK_at_end_preamble:n
__xeCJK_after_preamble:n
__xeCJK_after_end_preamble:n

在 \document 前后加上各种钩子。

```
\tl_new:N \g__xeCJK_at_end_preamble_hook_tl
\tl_new:N \g__xeCJK_after_preamble_hook_tl
\tl_new:N \g__xeCJK_after_end_preamble_hook_tl
\cs_new_protected:Npn \__xeCJK_at_end_preamble:n #1
  { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl {#1} }
\cs_new_protected:Npn \__xeCJK_after_preamble:n #1
  { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl {#1} }
\cs_new_protected:Npn \__xeCJK_after_end_preamble:n #1
  { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl {#1} }
\xeCJK_if_package_loaded:nTF { etoolbox }
  {
    \AtEndPreamble { \g__xeCJK_at_end_preamble_hook_tl }
    \AfterPreamble { \g__xeCJK_after_preamble_hook_tl }
    \AfterEndPreamble { \g__xeCJK_after_end_preamble_hook_tl }
  }
  {
    \AtBeginDocument { \g__xeCJK_after_preamble_hook_tl }
    \cs_new_protected_nopar:Npn \__xeCJK_document_left_hook:
      { \group_end: \g__xeCJK_at_end_preamble_hook_tl \group_begin: }
    \cs_new_protected_nopar:Npn \__xeCJK_document_right_hook:
      { \scan_stop: \g__xeCJK_after_end_preamble_hook_tl \tex_ignorespaces:D }
    \cs_gset_nopar:Npx \document
      {
        \__xeCJK_document_left_hook:
        \exp_not:o { \document }
        \__xeCJK_document_right_hook:
      }
  }
```

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

\xeCJKShipoutHook

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

```
\AddEverypageHook { \xeCJKShipoutHook }
\NewDocumentCommand \xeCJKShipoutHook { }
  {
    \bool_if:NF \l__xeCJK_shipout_hook_bool
      {
```

```
        \bool_set_true:N \l__xeCJK_shipout_hook_bool
        \tl_use:N \l__xeCJK_shipout_hook_tl
      }
  }
\cs_new_protected:Npn \__xeCJK_add_to_shipout:n #1
  { \tl_put_right:Nn \l__xeCJK_shipout_hook_tl {#1} }
\tl_new:N \l__xeCJK_shipout_hook_tl
\bool_new:N \l__xeCJK_shipout_hook_bool
```

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

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

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

(*End definition for* `\xeCJK_reverse:nnn`.)

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

```
\cs_new_protected_nopar:Npn \xeCJK_tl_remove_outer_braces:N #1
  { \tl_set:Nx #1 { \exp_args:NV \xeCJK_tl_remove_outer_braces:n #1 } }
\cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
  {
    \exp_last_unbraced:Nf
    \__xeCJK_tl_remove_outer_braces:w { \tl_trim_spaces:n {#1} } \s__stop
  }
\cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \s__stop
  {
    \bool_if:nTF { \tl_if_single_p:n {#1} && ! ( \tl_if_head_is_N_type_p:n {#1} ) }
      { \xeCJK_tl_remove_outer_braces:n {#1} }
      { \tl_trim_spaces:n {#1} }
  }
```

(*End definition for* `\xeCJK_tl_remove_outer_braces:N` *and* `\xeCJK_tl_remove_outer_braces:n`.)

`\xeCJK_cs_clear:N` `\xeCJK_cs_gclear:N` 让控制序列的意义为空。

```
\cs_new_protected:Npn \xeCJK_cs_clear:N #1
  { \cs_set_eq:NN #1 \prg_do_nothing: }
\cs_new_protected:Npn \xeCJK_cs_gclear:N #1
  { \cs_gset_eq:NN #1 \prg_do_nothing: }
```

(*End definition for* `\xeCJK_cs_clear:N` *and* `\xeCJK_cs_gclear:N`.)

`\xeCJK_swap_cs:NN` 交换 #1 和 #2 的意义。

```
\cs_new_protected:Npn \xeCJK_swap_cs:NN #1#2
  {
    \cs_set_eq:NN \__xeCJK_swap_cs_aux:w #1
    \cs_set_eq:NN #1 #2
    \cs_set_eq:NN #2 \__xeCJK_swap_cs_aux:w
    \cs_undefine:N \__xeCJK_swap_cs_aux:w
  }
```

(*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。

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

(*End definition for* `\xeCJK_font_gset_to_current:c`.)

`\xeCJK_glyph_if_exist_p:N` `\xeCJK_glyph_if_exist:NTF` 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。

```
\prg_new_conditional:Npnn \xeCJK_glyph_if_exist:N #1 { p , T , F , TF }
  {
    \etex_iffontchar:D \tex_font:D `#1 \exp_stop_f:
      \prg_return_true: \else: \prg_return_false: \fi:
  }
```

(*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。

```
\tl_const:Nn \c_xeCJK_space_skip_tl
  {
    \int_compare:nNnTF \g__xeCJK_spacefactor_int = \c_one_thousand
      {
        \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
          {
            \tex_fontdimen:D \c_two \tex_font:D
              plus  \tex_fontdimen:D \c_three \tex_font:D
              minus \tex_fontdimen:D \c_four  \tex_font:D
          }
          { \tex_spaceskip:D }
      }
      {
        \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
          {
            \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
             { \__xeCJK_space_skip_scale:nnn { \tex_fontdimen:D \c_two \tex_font:D } }
              {
                \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
                  {
                    \__xeCJK_space_skip_scale:nnn
                      {
                        \tex_fontdimen:D \c_two   \tex_font:D +
                        \tex_fontdimen:D \c_seven \tex_font:D
                      }
                  }
                  { \tex_xspaceskip:D  \use_none:nn }
              }
              { \tex_fontdimen:D \c_three \tex_font:D }
              { \tex_fontdimen:D \c_four  \tex_font:D }
          }
          {
            \int_compare:nNnTF \g__xeCJK_spacefactor_int < { 2000 }
              { \__xeCJK_space_skip_scale:nnn { \tex_spaceskip:D } }
              {
                \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
                  {
                    \__xeCJK_space_skip_scale:nnn
                      { \tex_spaceskip:D + \tex_fontdimen:D \c_seven \tex_font:D }
                  }
                  { \tex_xspaceskip:D \use_none:nn }
              }
              { \etex_gluestretch:D \tex_spaceskip:D }
              { \etex_glueshrink:D  \tex_spaceskip:D }
          }
      }
  }
\cs_new_nopar:Npn \__xeCJK_space_skip_scale:nnn #1#2#3
  {
    \dim_eval:n {#1}
    plus \fp_eval:n { \g__xeCJK_spacefactor_int / 1000 } #2
    minus
      \int_div_truncate:nn
        { 1000 * \tex_number:D #3 } { \g__xeCJK_spacefactor_int } sp
  }
\int_new:N \g__xeCJK_spacefactor_int
\int_gset_eq:NN \g__xeCJK_spacefactor_int \c_one_thousand
```

(*End definition for* \c_xeCJK_space_skip_tl.)

\xeCJK_glue_to_skip:nN 取得一个 glue 的长度,包括伸展和收缩部分。如果参数不是 glue,则取其宽度。

```
\cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2
  {
    \group_begin:
    \hbox_set:Nw \l__xeCJK_tmp_box #1 \scan_stop:
    \int_compare:nNnTF \etex_lastnodetype:D = \c_eleven
      {
        \exp_after:wN \hbox_set_end: \exp_after:wN \group_end: \exp_after:wN
```

```
        \skip_set:Nn \exp_after:wN #2 \exp_after:wN
          { \skip_use:N \tex_lastskip:D }
      }
      {
        \hbox_set_end: \exp_after:wN \group_end: \exp_after:wN
        \skip_set:Nn \exp_after:wN #2 \exp_after:wN
           { \dim_use:N \box_wd:N \l__xeCJK_tmp_box }
      }
  }
```

(*End definition for* \xeCJK_glue_to_skip:nN.)

\xeCJK_if_blank_x_p:n
\xeCJK_if_blank_x:nTF

判断是否为空或者仅含一个空格。

```
\prg_new_conditional:Npnn \xeCJK_if_blank_x:n #1 { p , T , F , TF }
  {
    \if_case:w \pdftex_strcmp:D { } {#1} \exp_stop_f:
      \prg_return_true:
    \else:
      \if_case:w \pdftex_strcmp:D { ~ } {#1} \exp_stop_f:
        \prg_return_true: \else: \prg_return_false: \fi:
    \fi:
  }
```

(*End definition for* \xeCJK_if_blank_x:n.)

\xeCJK_int_until_do:nn
__xeCJK_int_until_do:wn

由于定义较为简单,可以比 \int_until_do:nNnn 稍微快一点点。

```
\cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
  { \__xeCJK_int_until_do:wn \use_none:n { \reverse_if:N \if_int_compare:w #1#2 } }
\cs_new_protected:Npn \__xeCJK_int_until_do:wn \use_none:n #1
  { #1 \exp_after:wN \__xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
\int_new:N \l__xeCJK_begin_int
\int_new:N \l__xeCJK_end_int
```

(*End definition for* \xeCJK_int_until_do:nn *and* __xeCJK_int_until_do:wn.)

\xeCJK_peek_catcode_ignore_spaces:NTF

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

```
\cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
  {
    \cs_set_eq:NN \l__peek_search_token #1 \scan_stop:
    \tl_set:Nn \__xeCJK_peek_catcode_true:w  { \group_align_safe_end: #2 }
    \tl_set:Nn \__xeCJK_peek_catcode_false:w { \group_align_safe_end: #3 }
    \bool_set_false:N \l__xeCJK_peek_ignore_spaces_bool
    \group_align_safe_begin:
    \peek_after:Nw \__xeCJK_peek_catcode_ignore_spaces_branches:w
  }
\cs_new_protected_nopar:Npn \__xeCJK_peek_catcode_ignore_spaces_branches:w
  {
    \if_meaning:w \l_peek_token \c_space_token
      \bool_set_true:N \l__xeCJK_peek_ignore_spaces_bool
      \exp_after:wN \peek_after:Nw
      \exp_after:wN \__xeCJK_peek_catcode_ignore_spaces_branches:w
      \tex_romannumeral:D 0
    \else:
      \if_catcode:w
        \exp_not:N \l_peek_token \exp_not:N \l__peek_search_token
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__xeCJK_peek_catcode_true:w
      \else:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__xeCJK_peek_catcode_false:w
      \fi:
    \fi:
  }
\tl_new:N \__xeCJK_peek_catcode_true:w
\tl_new:N \__xeCJK_peek_catcode_false:w
\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。

```
\cs_new_protected:Npn \xeCJK_peek_after_ignore_spaces:nw #1
  {
    \tl_set:Nn \__xeCJK_peek_after_do:w { \group_align_safe_end: #1 }
    \group_align_safe_begin:
    \peek_after:Nw \__xeCJK_peek_ignore_spaces_branches:w
  }
\cs_new_protected_nopar:Npn \__xeCJK_peek_ignore_spaces_branches:w
  {
    \if_meaning:w \l_peek_token \c_space_token
      \exp_after:wN \peek_after:Nw
      \exp_after:wN \__xeCJK_peek_ignore_spaces_branches:w
      \tex_romannumeral:D 0
    \else:
      \exp_after:wN \__xeCJK_peek_after_do:w
    \fi:
  }
```

(*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。

```
\cs_new_nopar:Npn \xeCJK_token_value_class:N #1
  { \XeTeXcharclass \xeCJK_token_value_charcode:N #1 }
```

(*End definition for \xeCJK_token_value_class:N.*)

\xeCJK_token_value_charcode:N

当记号 #1 的 charcode 大于等于 0x10000 时，XeTeX 0.9999.0 版以前的 \meaning 的返回结果比较特殊[4]，需要特别处理。同时使用较新版本中提供的原语设置 mathcode。目前，0.9999.0 版以后的 XeTeX 的 \meaning 对于超出 BMP 的字符，会返回两个字符，分别对应于其 UTF-16 编码的首尾代理。[5]

```
\cs_new_nopar:Npn \xeCJK_token_value_charcode:N #1
  { \exp_after:wN \__xeCJK_token_value_charcode:w \token_to_meaning:N #1 \q_stop }
\fp_compare:nNnTF { \int_use:N \xetex_XeTeXversion:D \XeTeXrevision } > { 0.9998 }
  {
    \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
      {
        \int_eval:n
          {
            \tl_if_empty:nTF {#4}
              { `#3 }
              { ( `#3 - "D800 ) * "400 + ( `#4 - "DC00 ) + "10000 }
          }
      }
    \cs_new_eq:NN \xeCJK_xetex_mathcode:w \Umathcode
  }
  {
    \cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3#4 \q_stop
      { \int_eval:n { \tl_if_empty:nTF {#4} { `#3 } { "20000 } } }
    \cs_new_eq:NN \xeCJK_xetex_mathcode:w \XeTeXmathcode
  }
```

(*End definition for \xeCJK_token_value_charcode:N.*)

\xeCJK_if_CJK_class_p:N
\xeCJK_if_CJK_class:N*TF*

判断字符 #1 是否为 CJK 字符类，包括文字和标点符号。

```
\prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
  {
    \if_cs_exist:w \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 } \cs_end:
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\cs_new_nopar:Npn \__xeCJK_CJK_class_tl:n #1
  { c__xeCJK_CJK_class_ \int_eval:n {#1} _tl }
\cs_generate_variant:Nn \__xeCJK_CJK_class_tl:n { c }
```

(*End definition for \xeCJK_if_CJK_class:N.*)

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

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

\xeCJK_if_same_class_p:NN
\xeCJK_if_same_class:NNTF

判断两个字符是否同属于一个字符类。

```
\prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
  {
    \if_int_compare:w
      \xeCJK_token_value_class:N #1 = \xeCJK_token_value_class:N #2 \exp_stop_f:
      \prg_return_true: \else: \prg_return_false: \fi:
  }
```

(*End definition for* \xeCJK_if_same_class:NN.)

5.3 功能开关

xeCJKactive

事实上，将开启或关闭 XeTeX 的整个字符类机制。

```
\keys_define:nn { xeCJK / options }
  {
    xeCJKactive .choice: ,
    xeCJKactive / true  .code:n = { \makexeCJKactive   } ,
    xeCJKactive / false .code:n = { \makexeCJKinactive } ,
    xeCJKactive      .default:n = { true }
  }
```

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

\makexeCJKactive
\makexeCJKinactive

```
\NewDocumentCommand \makexeCJKactive   { } { \XeTeXinterchartokenstate = \c_one  }
\NewDocumentCommand \makexeCJKinactive { } { \XeTeXinterchartokenstate = \c_zero }
```

(*End definition for* \makexeCJKactive *and* \makexeCJKinactive.)

抑制 BOM。

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

5.4 字符类别设定

\g__xeCJK_class_seq
\g__xeCJK_new_class_seq

分别用于记录在 xeCJK 中使用的字符类别名称和新建的字符类别的编号。

```
\seq_new:N \g__xeCJK_class_seq
\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。

```
\cs_new_protected_nopar:Npn \xeCJK_new_class:n #1
  {
    \int_if_exist:cTF { \__xeCJK_class_csname:n {#1} }
      { \__xeCJK_error:nx { class-already-defined } {#1} }
      {
        \exp_args:Nc \newXeTeXintercharclass { \__xeCJK_class_csname:n {#1} }
        \clist_new:c { g__xeCJK_#1_range_clist }
        \seq_gput_right:Nn \g__xeCJK_class_seq {#1}
        \seq_gput_right:Nv \g__xeCJK_new_class_seq { \__xeCJK_class_csname:n {#1} }
      }
  }
```

(*End definition for* \xeCJK_new_class:n.)

\xeCJK_save_class:nn

保存 XeTeX 预定义的字符类别。#1 是自定义名称，#2 是编号。

```
\cs_new_protected_nopar:Npn \xeCJK_save_class:nn #1#2
  {
    \int_if_exist:cTF { \__xeCJK_class_csname:n {#1} }
      { \__xeCJK_error:nx { class-already-defined } {#1} }
      {
        \int_const:cn { \__xeCJK_class_csname:n {#1} } {#2}
        \clist_new:c { g__xeCJK_#1_range_clist }
        \seq_gput_right:Nn \g__xeCJK_class_seq {#1}
      }
  }
```

(*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。

```
\cs_new_nopar:Npn \__xeCJK_class_csname:n #1 { c__xeCJK_#1_class_int }
\cs_new_eq:cN { \__xeCJK_class_csname:n { Others } } \l__xeCJK_tmp_int
\__xeCJK_msg_new:nn { class-already-defined }
  {
    XeTeX~character~class~`#1'~has~been~already~defined.\\\\
    Please~take~another~name. \\
  }
```

(*End definition for* `\__xeCJK_class_csname:n`.)

xeCJK 需要以下字符类别用于字符输出。其中 `Default`、`CJK`、`FullLeft`、`FullRight`、`Boundary` 为 XeTeX 中预定义的类别，xeCJK 新增加了 `HalfLeft`、`HalfRight`、`NormalSpace` 和 `IVS`。其中异体字选择符 (Ideographic Variation Selectors)[6] 需要 XeTeX `0.9999.0` 以上的版本[7]和相关字体的支持。

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

`Default`, `CJK`, `FullLeft`, `FullRight`, `Boundary` 这五类是 XeTeX 预定义的类别。

```
\xeCJK_save_class:nn { Default }   { \c_zero  }
\xeCJK_save_class:nn { CJK }       { \c_one   }
\xeCJK_save_class:nn { FullLeft }  { \c_two   }
\xeCJK_save_class:nn { FullRight } { \c_three }
\xeCJK_save_class:nn { Boundary }  { \c_two_hundred_fifty_five }
```

(*End definition for* `Default` *and others.*)

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

```
\xeCJK_new_class:n { HalfLeft }
\xeCJK_new_class:n { HalfRight }
\xeCJK_new_class:n { NormalSpace }
\xeCJK_new_class:n { IVS }
```

(*End definition for* `HalfLeft` *and others.*)

`\c__xeCJK_HalfLeft_chars_clist`, `\c__xeCJK_HalfRight_chars_clist`, `\c__xeCJK_NormalSpace_chars_clist` 西文半角左/右标点和前后原始间距的字符类。

```
\clist_const:Nn \c__xeCJK_HalfLeft_chars_clist
  { "28 , "2D , "5B , "60 , "7B }
\clist_const:Nn \c__xeCJK_HalfRight_chars_clist
  { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D }
\clist_const:Nn \c__xeCJK_NormalSpace_chars_clist { "2F }
```

(*End definition for* `\c__xeCJK_HalfLeft_chars_clist`, `\c__xeCJK_HalfRight_chars_clist`, *and* `\c__xeCJK_NormalSpace_chars_c`

以下对全角标点符号的归类来源于 XeTeX 的脚本 `unicode-char-prep.pl` 和 Unicode 数据库[8]。

`\c__xeCJK_OP_chars_clist` Open Punctuation (OP)

U+2018	‘	U+201C	“	U+2329	〈	U+3008	〈	U+300A	《	U+300C	「	U+300E	『
U+3010	【	U+3014	〔	U+3016	〖	U+3018	〘	U+301A	〚	U+301D	〝	U+FE17	︗
U+FE35	︵	U+FE37	︷	U+FE39	︹	U+FE3B	︻	U+FE3D	︽	U+FE3F	︿	U+FE41	︁
U+FE43	︃	U+FE47	︇	U+FE59	﴿	U+FE5B	｛	U+FE5D	﴿	U+FF08	（	U+FF3B	［
U+FF5B	｛	U+FF5F	⦅	U+FF62	｢								

以下代码的第一行是中西文共用的左引号。

```
\clist_const:Nn \c__xeCJK_OP_chars_clist
```

[6] http://www.unicode.org/reports/tr37/
[7] http://tug.org/pipermail/xetex/2013-March/024118.html
[8] http://www.unicode.org/reports/tr14/

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

(*End definition for* `\c__xeCJK_OP_chars_clist`.)

`\c__xeCJK_PR_chars_clist` Prefix Numeric (PR)

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

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

(*End definition for* `\c__xeCJK_PR_chars_clist`.)

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

```
\clist_const:Nx \c__xeCJK_FullLeft_chars_clist
  {
    \c__xeCJK_OP_chars_clist ,
    \c__xeCJK_PR_chars_clist
  }
```

(*End definition for* `\c__xeCJK_FullLeft_chars_clist`.)

`\c__xeCJK_CL_chars_clist` Close Punctuation (CL)

U+00B7	·	U+2019	’	U+201D	”	U+2014	—	U+2015	―	U+2025	‥	U+2026	…
U+2027	‧	U+2500	─	U+232A	〉	U+3001	、	U+3002	。	U+3009	〉	U+300B	》
U+300D	」	U+300F	』	U+3011	】	U+3015	〕	U+3017	〗	U+3019	〙	U+301B	〛
U+301E	″	U+301F	‟	U+FE11	︐	U+FE12	︑	U+FE18	︘	U+FE36	︶	U+FE38	︸
U+FE3A	︺	U+FE3C	︼	U+FE3E	︾	U+FE40	︿	U+FE42	﹂	U+FE44	﹄	U+FE48	︾
U+FE50	，	U+FE52	．	U+FE5A	）	U+FE5C	｝	U+FE5E	｠	U+FF09	）	U+FF0C	，
U+FF0E	．	U+FF3D	］	U+FF5D	｝	U+FF60	〕	U+FF61	｡	U+FF63	」	U+FF64	、

以下代码的第一行是中西文共用的一些标点符号。

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

(*End definition for* `\c__xeCJK_CL_chars_clist`.)

`\c__xeCJK_NS_chars_clist` Nonstarter (NS)

U+3005	々	U+301C	〜	U+303B	〻	U+303C	〼	U+309B	゛	U+309C	゜	U+309D	ゝ
U+309E	ゞ	U+30A0	゠	U+30FB	・	U+30FD	ヽ	U+30FE	ヾ	U+A015	ꕸ	U+FE54	；
U+FE55	：	U+FF1A	：	U+FF1B	；	U+FF65	･	U+FF9E	ﾞ	U+FF9F	ﾟ		

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

(*End definition for* `\c__xeCJK_NS_chars_clist`.)

`\c__xeCJK_EX_chars_clist` Exclamation/Interrogation (EX)

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

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

(*End definition for* `\c__xeCJK_EX_chars_clist`.)

`\c__xeCJK_IS_chars_clist` Infix Numeric Separator (IS)

U+FE10	︐	U+FE13	︓	U+FE14	︔

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

(*End definition for* `\c__xeCJK_IS_chars_clist`.)

\c__xeCJK_CJ_chars_clist Conditional Japanese Starter (CJ)

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

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

(*End definition for* \c__xeCJK_CJ_chars_clist.)

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

U+FE6A	﹪	U+FF05	％	U+FFE0	￠

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

(*End definition for* \c__xeCJK_PO_chars_clist.)

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

```
\clist_const:Nx \c__xeCJK_FullRight_chars_clist
  {
    \c__xeCJK_CL_chars_clist ,
    \c__xeCJK_NS_chars_clist ,
    \c__xeCJK_EX_chars_clist ,
    \c__xeCJK_IS_chars_clist ,
    \c__xeCJK_CJ_chars_clist ,
    \c__xeCJK_PO_chars_clist
  }
```

(*End definition for* \c__xeCJK_FullRight_chars_clist.)

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

```
\clist_const:Nn \c__xeCJK_CJK_chars_clist
  {
```

- Hangul Jamo (谚文字母)

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

- CJK Radicals Supplement (中日韩部首补充)

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

- Kangxi Radicals (康熙部首)

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

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

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

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

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

- Hiragana (日文平假名)

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

- Katakana (日文片假名)

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

- Bopomofo (注音字母)

```
"3100 -> "312F ,
```

- Hangul Compatibility Jamo (谚文兼容字母)

```
"3130 -> "318F ,
```

- Kanbun (象形字注释标志)

```
"3190 -> "319F ,
```

- Bopomofo Extended (注音字母扩展)

```
"31A0 -> "31BF ,
```

- CJK Strokes (中日韩笔画)

```
"31C0 -> "31EF ,
```

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

```
"31F0 -> "31FF ,
```

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

```
"3200 -> "32FF ,
```

- CJK Compatibility (中日韩兼容)

```
"3300 -> "33FF ,
```

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

```
"3400 -> "4DBF ,
```

- Yijing Hexagrams Symbols (易经六十四卦符号)

```
"4DC0 -> "4DFF ,
```

- CJK Unified Ideographs (中日韩统一表意文字)

```
"4E00 -> "9FFF ,
```

- Yi Syllables (彝文音节)

```
"A000 -> "A48F ,
```

- Yi Radicals (彝文字根)

```
"A490 -> "A4CF ,
```

- Hangul Jamo Extended-A (谚文扩展 A)

```
"A960 -> "A97F ,
```

- Hangul Syllables (谚文音节)

```
"AC00 -> "D7AF ,
```

- Hangul Jamo Extended-B (谚文扩展 B)

```
"D7B0 -> "D7FF ,
```

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

```
"F900 -> "FAFF ,
```

- Vertical Forms (竖排形式)

```
"FE10 -> "FE1F ,
```

- CJK Compatibility Forms (中日韩兼容形式)

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

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

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

- Kana Supplement (日文假名增补)

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

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

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

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

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

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

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

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

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

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

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

(*End definition for* `\c__xeCJK_CJK_chars_clist`.)

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

```
\clist_const:Nn \c__xeCJK_IVS_chars_clist
  {
```

- 日文假名浊点

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

- Variation Selectors (异体字选择符)

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

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

```
    "E0100 -> "E01EF
  }
```

(*End definition for* `\c__xeCJK_IVS_chars_clist`.)

5.5 字符类别处理

`\g__xeCJK_base_class_seq`
`\g__xeCJK_non_CJK_class_seq`
`\g__xeCJK_CJK_class_seq`

```
\seq_new:N \g__xeCJK_base_class_seq
\seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq
\seq_new:N \g__xeCJK_non_CJK_class_seq
\seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq
  { Default , HalfLeft , HalfRight , NormalSpace , Boundary }
\seq_new:N \g__xeCJK_CJK_class_seq
\cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
  {
    \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
    \tl_const:cn { \__xeCJK_CJK_class_tl:c { \__xeCJK_class_csname:n {#1} } } {#1}
  }
\clist_map_function:nN { CJK , FullLeft , FullRight , IVS } \__xeCJK_save_CJK_class:n
```

(*End definition for* `\g__xeCJK_base_class_seq`, `\g__xeCJK_non_CJK_class_seq`, *and* `\g__xeCJK_CJK_class_seq`.)

`\xeCJK_class_num:n` #1 为字符类别名称,用于取得字符类别对应的编号。

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

(*End definition for* `\xeCJK_class_num:n`.)

\xeCJKDeclareCharClass

```
\NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
  {
    \xeCJK_declare_char_class:nx {#2} {#3}
    \IfBooleanT {#1} { \xeCJKResetPunctClass }
  }
```

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

xeCJK_declare_char_class:nn
__xeCJK_set_char_class_aux:Nnw

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。

```
\cs_new_protected_nopar:Npn \xeCJK_declare_char_class:nn #1#2
  {
    \clist_set:Nn \l__xeCJK_tmp_clist {#2}
    \clist_gconcat:ccN
      { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmp_clist
    \clist_map_inline:Nn \l__xeCJK_tmp_clist
      {
        \str_if_eq:nnF {##1} { -> }
          {
            \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
              { \xeCJK_class_num:n {#1} }
          }
      }
    \xeCJK_set_char_class:nnn { "3099 } { "309A } { \xeCJK_class_num:n { IVS } }
  }
\NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw
  { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
\cs_generate_variant:Nn \clist_gconcat:NNN { cc }
\cs_generate_variant:Nn \xeCJK_declare_char_class:nn { nx , nV }
```

(*End definition for* \xeCJK_declare_char_class:nn *and* __xeCJK_set_char_class_aux:Nnw.)

__xeCJK_check_num_range:nnNN

```
\cs_new_protected_nopar:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
  {
    \bool_if:nTF { \xeCJK_if_blank_x_p:n {#1} || \xeCJK_if_blank_x_p:n {#2} }
      {
        \int_set:Nn #3 { \xeCJK_if_blank_x:nTF {#1} {#2} {#1} }
        \int_set_eq:NN #3 #4
      }
      {
        \int_set:Nn #3 { \int_min:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
        \int_set:Nn #4 { \int_max:nn {#1} { \IfNoValueTF {#2} {#1} {#2} } }
      }
  }
```

(*End definition for* __xeCJK_check_num_range:nnNN.)

```
\int_set:Nn \l__xeCJK_begin_int { "AC00 }
\int_set:Nn \l__xeCJK_end_int   { "D7A3 }
\xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
  {
    \char_set_catcode_letter:n { \l__xeCJK_begin_int }
    \int_incr:N \l__xeCJK_begin_int
  }
```

\xeCJK_set_char_class:nnn

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

```
\cs_new_protected_nopar:Npn \xeCJK_set_char_class:nnn #1#2#3
  {
    \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
    \int_set:Nn \l__xeCJK_tmp_int {#3}
    \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
      {
        \XeTeXcharclass \l__xeCJK_begin_int = \l__xeCJK_tmp_int
        \int_incr:N \l__xeCJK_begin_int
      }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
  {
    \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#2} }
    \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
      { \XeTeXcharclass ##1 = \l__xeCJK_tmp_int }
  }
```

(*End definition for* __xeCJK_set_char_class_eq:nn.)

\normalspacedchars 声明前后不加间距的字符。

```
\NewDocumentCommand \normalspacedchars { m }
  {
    \tl_map_inline:nn {#1}
      { \XeTeXcharclass `##1 = \xeCJK_class_num:n { NormalSpace } }
  }
```

(*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。

```
\NewDocumentCommand \xeCJKResetPunctClass { }
  {
    \xeCJK_declare_char_class:nV { HalfLeft  } \c__xeCJK_HalfLeft_chars_clist
    \xeCJK_declare_char_class:nV { HalfRight } \c__xeCJK_HalfRight_chars_clist
    \xeCJK_declare_char_class:nV { FullLeft  } \c__xeCJK_FullLeft_chars_clist
    \xeCJK_declare_char_class:nV { FullRight } \c__xeCJK_FullRight_chars_clist
  }
```

(*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。

```
\NewDocumentCommand \xeCJKResetCharClass { }
  {
    \xeCJK_declare_char_class:nV { CJK } \c__xeCJK_CJK_chars_clist
    \xeCJK_declare_char_class:nV { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
    \xeCJK_declare_char_class:nV { IVS } \c__xeCJK_IVS_chars_clist
    \xeCJKResetPunctClass
  }
```

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

设置字符类别。

```
\xeCJKResetCharClass
```

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

```
\cs_new_protected_nopar:Npn \xeCJK_inter_class_toks:nnn #1#2#3
  { \XeTeXinterchartoks \xeCJK_class_num:n {#1} ~ \xeCJK_class_num:n {#2} = {#3} }
\cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nnc , nnx }
```

(*End definition for* \xeCJK_inter_class_toks:nnn.)

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

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

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

```
\cs_new_protected_nopar:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
  {
    \xeCJK_inter_class_toks:nnx {#1} {#2}
      { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
  }
\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。

```
\cs_new_protected_nopar:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
  {
    \xeCJK_inter_class_toks:nnx {#1} {#2}
      { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
  }
\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。

```
\cs_new_protected_nopar:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
  {
    \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
    \tl_if_empty:NF \l__xeCJK_tmp_tl
      { \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmp_tl } }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
  {
    \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
    \tl_if_empty:NF \l__xeCJK_tmp_tl
      {
        \tl_replace_all:Nnn \l__xeCJK_tmp_tl {#3} {#4}
        \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmp_tl }
      }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_clear_Boundary_and_CJK_toks:
 { \seq_map_function:NN \g__xeCJK_CJK_class_seq \__xeCJK_clear_Boundary_and_CJK_toks:n }
\cs_new_protected_nopar:Npn \__xeCJK_clear_Boundary_and_CJK_toks:n #1
  { \xeCJK_clear_inter_class_toks:nn { Boundary } {#1} }
```

(*End definition for* \xeCJK_clear_Boundary_and_CJK_toks:.)

5.6 字符输出规则

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

\xeCJK_class_group_begin:
\xeCJK_class_group_end:

```
\cs_new_protected_nopar:Npn \xeCJK_class_group_begin:
  {
    \c_group_begin_token
    \bool_set_true:N \l__xeCJK_CJK_group_bool
  }
\bool_new:N \l__xeCJK_CJK_group_bool
\cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token
```

(*End definition for* \xeCJK_class_group_begin: *and* \xeCJK_class_group_end:.)

IVS 字符类与 CJK 字符类基本相同,只是从 CJK 转移到 IVS 时,不加入任何内容。

```
\AtEndOfPackage
  {
    \seq_map_inline:Nn \g__xeCJK_class_seq
      {
        \str_if_eq:nnTF {#1} { IVS }
          { \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } { CJK } }
          {
            \xeCJK_copy_inter_class_toks:nnnn { IVS } {#1} { CJK } {#1}
            \str_if_eq:nnF {#1} { CJK }
              { \xeCJK_copy_inter_class_toks:nnnn {#1} { IVS } {#1} { CJK } }
          }
      }
  }
\clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
  {
    \xeCJK_inter_class_toks:nnn {#1} { CJK }
      {
        \xeCJK_class_group_begin:
        \xeCJK_select_font:
        \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
        \xeCJK_clear_Boundary_and_CJK_toks:
        \CJKsymbol
      }
    \xeCJK_inter_class_toks:nnn { CJK } {#1} { \xeCJK_class_group_end: }
  }
\clist_map_inline:nn { Default , HalfLeft }
  {
    \xeCJK_inter_class_toks:nnn { Boundary } {#1} { \xeCJK_Boundary_and_Default: }
    \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue }
  }
```

xeCJK_Boundary_and_Default:

```
\cs_new_protected_nopar:Npn \xeCJK_Boundary_and_Default:
  {
    \bool_if:nTF
      {
        \l__xeCJK_xecglue_bool &&
        \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
        \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
      }
      {
        \tex_unskip:D
        \bool_if:nTF
          {
            \xeCJK_if_last_node_p:n { CJK }       ||
            \xeCJK_if_last_node_p:n { CJK-space }
          }
          { \xeCJK_remove_node: \CJKecglue } { ~ }
      }
      {
        \bool_if:nTF
          {
            \xeCJK_if_last_node_p:n { CJK }         ||
            \xeCJK_if_last_node_p:n { CJK-nobreak }
          }
          { \xeCJK_remove_node: \CJKecglue }
          {
            \xeCJK_if_last_node:nT { CJK-space }
              { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
          }
      }
  }
```

(*End definition for* \xeCJK_Boundary_and_Default:.)

```
\clist_map_inline:nn { Default , HalfRight }
  {
```

```
    \xeCJK_inter_class_toks:nnn {#1} { Boundary }
      {
        \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
        \peek_meaning_remove:NTF \tex_italiccorrection:D
          { \tex_italiccorrection:D { \xeCJK_make_node:n { default } } }
          {
            \token_if_space:NTF \l_peek_token
              { { \xeCJK_make_node:n { default-space } } }
              { { \xeCJK_make_node:n { default } } }
          }
      }
    \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue }
  }

\xeCJK_inter_class_toks:nnn { Boundary } { NormalSpace }
  { \xeCJK_Boundary_and_NormalSp: }
```

\xeCJK_Boundary_and_NormalSp:

```
\cs_new_protected_nopar:Npn \xeCJK_Boundary_and_NormalSp:
  {
    \bool_if:nTF
      {
        \l__xeCJK_xecglue_bool &&
        \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
        \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl
      }
      {
        \tex_unskip:D
        \bool_if:nTF
          {
            \xeCJK_if_last_node_p:n { CJK }       ||
            \xeCJK_if_last_node_p:n { CJK-space }
          }
          { \xeCJK_remove_node: \CJKecglue } { ~ }
      }
      {
        \xeCJK_if_last_node:nT { CJK-space }
          { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
      }
  }
```

(*End definition for* *xeCJK_Boundary_and_NormalSp:*.)

```
\xeCJK_inter_class_toks:nnn { NormalSpace } { Boundary }
  {
    \int_gset_eq:NN \g__xeCJK_spacefactor_int \tex_spacefactor:D
    \peek_meaning_remove:NTF \tex_italiccorrection:D
      { \tex_italiccorrection:D { \xeCJK_make_node:n { normalspace } } }
      {
        \token_if_space:NTF \l_peek_token
          { { \xeCJK_make_node:n { default-space } } }
          { { \xeCJK_make_node:n { normalspace } } }
      }
  }
\xeCJK_inter_class_toks:nnn { Boundary } { CJK }
  {
    \xeCJK_check_for_glue:
    \xeCJK_class_group_begin:
    \xeCJK_clear_Boundary_and_CJK_toks:
    \xeCJK_select_font:
    \CJKsymbol
  }
```

\xeCJK_check_for_glue:

```
\cs_new_protected_nopar:Npn \xeCJK_check_for_glue:
  {
    \bool_if:nTF
      { \xeCJK_if_last_node_p:n { CJK } || \xeCJK_if_last_node_p:n { CJK-space } }
      { \xeCJK_remove_node: \CJKglue }
```

```
      {
        \xeCJK_if_last_node:nTF { CJK-nobreak }
          { \xeCJK_remove_node: \xeCJK_no_break: \CJKglue }
          {
            \bool_if:nTF
              {
                \xeCJK_if_last_node_p:n { default }              ||
                \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
              }
              { \xeCJK_remove_node: \CJKecglue }
              {
                \bool_if:nT
                  {
                    \l__xeCJK_xecglue_bool &&
                    \int_compare_p:nNn \etex_lastnodetype:D = \c_eleven &&
                    ( \skip_if_eq_p:nn \tex_lastskip:D \c_xeCJK_space_skip_tl ||
                      \skip_if_eq_p:nn \tex_lastskip:D \l__xeCJK_ecglue_skip )
                  }
                  {
                    \tex_unskip:D
                    \bool_if:nTF
                      {
                        \xeCJK_if_last_node_p:n { default-space }        ||
                        \int_compare_p:nNn \etex_lastnodetype:D = \c_ten ||
                        \xeCJK_if_last_node_p:n { default }
                      }
                      { \xeCJK_remove_node: \CJKecglue }
                      {
                        \bool_if:nTF
                          {
                            \xeCJK_if_last_node_p:n { CJK }       ||
                            \xeCJK_if_last_node_p:n { CJK-space }
                          }
                          {
                            \xeCJK_remove_node:
                            \bool_if:NTF \l__xeCJK_reserve_space_bool
                              { ~ } { \CJKglue }
                          }
                          { ~ }
                      }
                  }
              }
          }
      }
  }
```

(*End definition for* `\xeCJK_check_for_glue:`.)

`\xeCJK_if_last_node_p:n`
`\xeCJK_if_last_node:nTF`

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

(*End definition for* `\xeCJK_if_last_node:n`.)

`\xeCJK_def_node:nn`
`\xeCJK_make_node:n`

用于判断插入的各种 kern。

```
\cs_new_protected_nopar:Npn \xeCJK_def_node:nn #1#2
  {
    \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
      { \dim_gset:cn } { \dim_const:cn }
      { c__xeCJK_#1_node_dim } {#2}
  }
\cs_new_protected_nopar:Npn \xeCJK_make_node:n #1
  {
    \tex_kern:D - \use:c { c__xeCJK_#1_node_dim }
    \tex_kern:D   \use:c { c__xeCJK_#1_node_dim }
  }
\cs_new_protected_nopar:Npn \xeCJK_remove_node:
```

```
  { \tex_unkern:D \tex_unkern:D }
\xeCJK_def_node:nn { CJK }           { 11 sp }
\xeCJK_def_node:nn { CJK-space }     { 12 sp }
\xeCJK_def_node:nn { default }       { 13 sp }
\xeCJK_def_node:nn { default-space } { 14 sp }
\xeCJK_def_node:nn { CJK-nobreak }   { 15 sp }
\xeCJK_def_node:nn { normalspace }   { 16 sp }
```

(*End definition for* `\xeCJK_def_node:nn` *and* `\xeCJK_make_node:n`.)

CJKglue CJK 文字之间插入的 glue。

```
\keys_define:nn { xeCJK / options }
  {
    CJKglue .code:n =
      {
        \cs_set_protected_nopar:Npn \CJKglue {#1}
        \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
      }
  }
\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

```
\keys_define:nn { xeCJK / options }
  {
    CJKecglue            .code:n =
      {
        \cs_set_protected_nopar:Npn \CJKecglue {#1}
        \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
      } ,
    xCJKecglue .choice: ,
    xCJKecglue / true    .code:n =
      {
        \bool_set_true:N  \l__xeCJK_xecglue_bool
        \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
      } ,
    xCJKecglue / false   .code:n =
      {
        \bool_set_false:N \l__xeCJK_xecglue_bool
        \cs_set_eq:NN \xeCJK_space_or_xecglue: \c_space_tl
      } ,
    xCJKecglue / unknown .code:n =
      {
        \bool_set_true:N  \l__xeCJK_xecglue_bool
        \cs_set_protected_nopar:Npn \CJKecglue {#1}
        \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
        \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
      } ,
    xCJKecglue        .default:n = { true }
  }
\skip_new:N \l__xeCJK_ecglue_skip
\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。

```
\keys_define:nn { xeCJK / options }
  {
    CJKspace .bool_set:N = \l__xeCJK_reserve_space_bool ,
    space        .meta:n = { CJKspace = true  } ,
    nospace      .meta:n = { CJKspace = false }
  }
```

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

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

\xeCJK_CJK_and_Boundary:w

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

```
\cs_new_protected_nopar:Npn \xeCJK_CJK_and_Boundary:w
  {
    \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
      {
        \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
          { \xeCJK_class_group_end: \xeCJK_space_or_xecglue: }
          { \xeCJK_class_group_end: \CJKecglue }
      }
      {
        \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
          {
            \bool_if:nTF
              {
                \token_if_macro_p:N \l_peek_token ||
               ( \l__xeCJK_reserve_space_bool && \token_if_letter_p:N \l_peek_token )
              }
              {
                \xeCJK_class_group_end: { \xeCJK_make_node:n { CJK-space } }
                \xeCJK_space_or_xecglue:
              }
              { \xeCJK_class_group_end: { \xeCJK_make_node:n { CJK-space } } }
          }
          {
            \token_if_eq_meaning:NNTF \l_peek_token \scan_stop:
              { \__xeCJK_CJK_and_Boundary_relax:N }
              { \__xeCJK_CJK_and_Boundary_aux: }
          }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_CJK_and_Boundary_aux:
  { \xeCJK_class_group_end: { \xeCJK_make_node:n { CJK } } }
\cs_new_protected:Npn \__xeCJK_CJK_and_Boundary_relax:N #1
  {
    \__xeCJK_CJK_and_Boundary_aux:
    \token_if_eq_meaning:NNTF #1 \scan_stop:
      {#1} { \cs_set_eq:NN #1 \scan_stop: #1 }
  }
```

(*End definition for \xeCJK_CJK_and_Boundary:w.*)

\xeCJK_ignore_spaces:w

```
\cs_new_protected_nopar:Npn \xeCJK_ignore_spaces:w
  {
    \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
      {
        \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
          { \xeCJK_space_or_xecglue: } { \CJKecglue }
      }
      {
        \bool_if:NT \l__xeCJK_peek_ignore_spaces_bool
          {
            \xeCJK_if_last_node:nT { CJK }
              { \xeCJK_remove_node: { \xeCJK_make_node:n { CJK-space } } }
            \bool_if:nT
              {
                \token_if_macro_p:N \l_peek_token ||
               ( \l__xeCJK_reserve_space_bool && \token_if_letter_p:N \l_peek_token )
              }
              { \xeCJK_space_or_xecglue: }
          }
      }
  }
```

(*End definition for \xeCJK_ignore_spaces:w.*)

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

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

\xeCJK_CJK_and_CJK:N

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

(*End definition for* \xeCJK_CJK_and_CJK:N.)

```
\xeCJK_inter_class_toks:nnn { FullLeft }  { CJK }
  { \xeCJK_FullLeft_and_CJK: \CJKsymbol }
\xeCJK_inter_class_toks:nnn { FullRight } { CJK }
  { \xeCJK_FullRight_and_CJK: \CJKsymbol }
\seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
  {
    \clist_map_inline:nn { FullLeft , FullRight }
      {
        \xeCJK_inter_class_toks:nnx {#1} {##1}
          { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
        \xeCJK_inter_class_toks:nnc {##1} {#1} { xeCJK_##1_and_Default: }
      }
  }
```

```
\xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
  { \xeCJK_Boundary_and_FullLeft:N  }
\xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
  { \xeCJK_Boundary_and_FullRight:N  }
```

\xeCJK_FullRight_and_Boundary:

```
\xeCJK_app_inter_class_toks:nnn { FullLeft } { Boundary } { \tex_ignorespaces:D }
\xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
  { \xeCJK_FullRight_and_Boundary: }
```

(*End definition for* \xeCJK_FullRight_and_Boundary:.)

\xeCJK_FullRight_and_Boundary:

```
\cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Boundary:
  { \xeCJK_FullRight_and_Default: \tex_ignorespaces:D }
```

(*End definition for* \xeCJK_FullRight_and_Boundary:.)

```
\clist_map_inline:nn { CJK , FullLeft , FullRight }
  {
    \clist_map_inline:nn { FullLeft , FullRight }
      { \xeCJK_inter_class_toks:nnc {#1} {##1} { xeCJK_#1_and_##1:N } }
  }
```

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

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_bound_rule:NN #1#2
  {
    \tex_vrule:D
      width - \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2}
      depth  \c_zero_dim
      height \c_zero_dim \scan_stop:
  }
```

(*End definition for* __xeCJK_punct_bound_rule:NN.)

__xeCJK_punct_rule:NN 用于减少标点符号的左/右空白。

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_rule:NN #1#2
  {
    \tex_vrule:D
      width  \__xeCJK_use_punct_dim:nnn { rule } {#1} {#2}
      depth  \c_zero_dim
      height \c_zero_dim \scan_stop:
  }
```

(*End definition for* __xeCJK_punct_rule:NN.)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_offset:NN #1#2
  { \__xeCJK_punct_kern:n { - \__xeCJK_use_punct_dim:nnn { offset } {#1} {#2} } }
\cs_new_protected_nopar:Npn \__xeCJK_punct_kern:n #1
  { \tex_kern:D #1 \exp_stop_f: }
```

(*End definition for* __xeCJK_punct_offset:NN.)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_glue:NN #1#2
  {
    \__xeCJK_punct_hskip:n
      {
        \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
        plus  \__xeCJK_use_punct_dim:nnn { plus } {#1} {#2}
        minus \__xeCJK_use_punct_dim:nnn { minus } {#1} {#2}
      }
  }
\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。

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

(*End definition for* __xeCJK_punct_kern:NN.)

\g__xeCJK_last_punct_tl 用于记录当前的标点符号。

```
\tl_new:N \g__xeCJK_last_punct_tl
```

(*End definition for* \g__xeCJK_last_punct_tl.)

\xeCJK_FullLeft_and_CJK:

```
\cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_CJK:
  {
    \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
      {
        \__xeCJK_punct_bound_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \xeCJK_no_break:
        \__xeCJK_punct_glue:NN \c__xeCJK_left_tl  \g__xeCJK_last_punct_tl
      }
      { \xeCJK_no_break: }
  }
```

(*End definition for* \xeCJK_FullLeft_and_CJK:.)

xeCJK_FullLeft_and_Default:

```
\cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_Default:
  {
    \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
      {
        \__xeCJK_punct_bound_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \xeCJK_class_group_end: \xeCJK_no_break:
        \__xeCJK_punct_glue:NN \c__xeCJK_left_tl  \g__xeCJK_last_punct_tl
      }
      { \xeCJK_class_group_end: \xeCJK_no_break: \__xeCJK_zero_glue: }
  }
\cs_new_protected_nopar:Npn \__xeCJK_zero_glue:
  { \skip_horizontal:N \c_zero_skip }
```

(*End definition for* \xeCJK_FullLeft_and_Default:.)

\xeCJK_FullRight_and_CJK:

```
\cs_new_protected_nopar:Npn \xeCJK_FullRight_and_CJK:
  {
    \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \CJKglue
  }
```

(*End definition for* \xeCJK_FullRight_and_CJK:.)

\xeCJK_FullRight_and_Default:

```
\cs_new_protected_nopar:Npn \xeCJK_FullRight_and_Default:
  {
    \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \xeCJK_class_group_end:
    \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
  }
```

(*End definition for* `\xeCJK_FullRight_and_Default:`.)

\xeCJK_Default_and_FullLeft:nN

```
\cs_new_protected_nopar:Npn \xeCJK_Default_and_FullLeft:nN #1#2
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#2}
    \__xeCJK_Default_and_FullLeft_glue:N {#2}
    \xeCJK_class_group_begin:
    \xeCJK_select_font:
    \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
    \xeCJK_clear_Boundary_and_CJK_toks:
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
    \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#2}
    \CJKpunctsymbol {#2}
  }
\cs_new_protected_nopar:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
  {
    \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
    \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
  }
```

(*End definition for* `\xeCJK_Default_and_FullLeft:nN`.)

\xeCJK_CJK_and_FullLeft:N

```
\cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullLeft:N #1
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
    \__xeCJK_CJK_and_FullLeft_glue:N {#1}
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
    \CJKpunctsymbol {#1}
  }
\cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
  {
    \CJKglue
    \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
    \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
  }
```

(*End definition for* `\xeCJK_CJK_and_FullLeft:N`.)

\xeCJK_Boundary_and_FullLeft:N

```
\cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullLeft:N #1
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
    \__xeCJK_Boundary_and_FullLeft_glue:N {#1}
    \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
    \xeCJK_class_group_begin:
    \xeCJK_select_font:
    \xeCJK_clear_Boundary_and_CJK_toks:
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \__xeCJK_punct_rule:NN \c__xeCJK_left_tl {#1}
    \CJKpunctsymbol {#1}
  }
```

(*End definition for* `\xeCJK_Boundary_and_FullLeft:N`.)

__xeCJK_Boundary_and_FullLeft_glue:N

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

```
\cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
```

```
  {
    \int_case:nnTF { \etex_lastnodetype:D }
      {
        { \c_one       }
        {
          \box_set_to_last:N \l__xeCJK_tmp_box
          \bool_if:nTF
            {
              \int_compare_p:nNn \etex_lastnodetype:D = \c_minus_one &&
              \dim_compare_p:nNn { \box_wd:N \l__xeCJK_tmp_box } = \tex_parindent:D
            }
            { \box_use_clear:N \l__xeCJK_tmp_box \use_none:n }
            { \box_use_clear:N \l__xeCJK_tmp_box \use:n }
        }
        { \c_minus_one } { \__xeCJK_zero_glue: \use_none:n }
        { \c_eleven    }
        {
          \bool_if:nTF
            {
              ! ( \skip_if_finite_p:n { \tex_lastskip:D } ) ||
              \skip_if_eq_p:nn { \tex_lastskip:D } { 1 sp }
            }
            { \__xeCJK_zero_glue: \use_none:n }
            {
              \skip_if_eq:nnTF { \tex_lastskip:D } { \labelsep }
                {
                  \tex_unskip:D
                  \bool_if:nTF
                    {
                      \int_compare_p:nNn \etex_lastnodetype:D = \c_thirteen &&
                      \int_compare_p:nNn \tex_lastpenalty:D = \c_zero
                    }
                    { \skip_horizontal:n { \labelsep } \use_none:n }
                    { \skip_horizontal:n { \labelsep } \use:n }
                }
                { \use:n }
            }
        }
        { \c_thirteen  }
        {
          \int_compare:nNnTF \tex_lastpenalty:D = \c_zero
            {
              \tex_unpenalty:D
              \int_compare:nNnTF \etex_lastnodetype:D = \c_one
                { \tex_penalty:D \c_zero \use_none:n }
                { \tex_penalty:D \c_zero \use:n }
            }
            { \use:n }
        }
      }
      { { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} } }
      { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
  }
```

(*End definition for* `\__xeCJK_Boundary_and_FullLeft_glue:N`.)

`\xeCJK_Default_and_FullRight:nN`

```
\cs_new_protected_nopar:Npn \xeCJK_Default_and_FullRight:nN #1#2
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#2}
    \__xeCJK_Default_and_FullRight_glue:N {#2}
    \xeCJK_class_group_begin:
    \xeCJK_select_font:
    \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
    \xeCJK_clear_Boundary_and_CJK_toks:
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#2}
    \xeCJK_FullRight_symbol:N {#2}
  }
```

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

\xeCJK_Boundary_and_FullRight:N

```
\cs_new_protected_nopar:Npn \xeCJK_Boundary_and_FullRight:N #1
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
    \__xeCJK_Default_and_FullRight_glue:N {#1}
    \xeCJK_class_group_begin:
    \xeCJK_select_font:
    \xeCJK_clear_Boundary_and_CJK_toks:
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \xeCJK_FullRight_symbol:N {#1}
  }
```

(*End definition for* \xeCJK_Boundary_and_FullRight:N.)

\xeCJK_CJK_and_FullRight:N

```
\cs_new_protected_nopar:Npn \xeCJK_CJK_and_FullRight:N #1
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
    \__xeCJK_CJK_and_FullRight_glue:N {#1}
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \xeCJK_FullRight_symbol:N {#1}
  }
```

(*End definition for* \xeCJK_CJK_and_FullRight:N.)

__xeCJK_CJK_and_FullRight_glue:N
__xeCJK_Default_and_FullRight_glue:N

```
\cs_new_protected_nopar:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
  {
    \__xeCJK_punct_if_long:NTF {#1}
      { \CJKglue }
      {
        \__xeCJK_punct_if_middle:NTF {#1}
          {
            \xeCJK_no_break:
            \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
            \__xeCJK_punct_bound_rule:NN \c__xeCJK_left_tl {#1}
          }
          { \xeCJK_no_break: }
      }
  }
\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

```
\cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullLeft:N #1
  {
    \xeCJK_no_break:
    \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
    \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
    \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \CJKpunctsymbol {#1}
  }
```

(*End definition for* \xeCJK_FullLeft_and_FullLeft:N.)

\xeCJK_FullLeft_and_FullRight:N

```
\cs_new_protected_nopar:Npn \xeCJK_FullLeft_and_FullRight:N #1
  {
    \xeCJK_no_break:
    \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
    \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
    \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \xeCJK_no_break:
    \xeCJK_FullRight_symbol:N {#1}
  }
```

(*End definition for* \xeCJK_FullLeft_and_FullRight:N.)

\xeCJK_FullRight_and_FullLeft:N

```
\cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullLeft:N #1
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl {#1}
    \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
    \xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \CJKpunctsymbol {#1}
  }
```

(*End definition for* \xeCJK_FullRight_and_FullLeft:N.)

__xeCJK_punct_nobreak_kern:NN

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_nobreak_kern:NN #1#2
  {
    \__xeCJK_punct_kern:NN #1#2
    \xeCJK_no_break:
  }
\cs_new_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_nobreak_kern:NN
```

(*End definition for* __xeCJK_punct_nobreak_kern:NN.)

__xeCJK_punct_breakable_kern:NN

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_breakable_kern:NN #1#2
  {
    \__xeCJK_punct_rule:NN \c__xeCJK_right_tl #1
    \__xeCJK_punct_breakable_kern:n
      { \__xeCJK_use_punct_dim:nnn { break_kern } {#1} {#2} }
    \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #2
  }
\cs_new_eq:NN \__xeCJK_punct_breakable_kern:n \skip_horizontal:n
```

(*End definition for* __xeCJK_punct_breakable_kern:NN.)

\xeCJK_FullRight_and_FullRight:N

```
\cs_new_protected_nopar:Npn \xeCJK_FullRight_and_FullRight:N #1
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl {#1}
    \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl {#1}
    \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl {#1}
    \tl_gset:Nx \g__xeCJK_last_punct_tl {#1}
    \xeCJK_no_break:
    \xeCJK_FullRight_symbol:N {#1}
  }
```

(*End definition for* \xeCJK_FullRight_and_FullRight:N.)

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```
\keys_define:nn { xeCJK / options }
  {
    CheckFullRight .choice: ,
    CheckFullRight / true  .code:n =
      {
        \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
          {
           \cs_set_eq:NN \__xeCJK_save_FullRight_check: \xeCJK_FullRight_and_Boundary:
             \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N \xeCJK_FullRight_symbol:N
             \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
             \cs_set_eq:NN \xeCJK_FullRight_symbol:N \xeCJK_check_FullRight_symbol:Nw
          }
      } ,
    CheckFullRight / false .code:n =
      {
        \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
          {
           \cs_set_eq:NN \xeCJK_FullRight_and_Boundary: \__xeCJK_save_FullRight_check:
             \cs_set_eq:NN \xeCJK_FullRight_symbol:N \__xeCJK_save_FullRight_symbol:N
          }
```

```
      } ,
    CheckFullRight       .default:n = { true }
  }
```

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

\xeCJK_FullRight_symbol:N

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

(*End definition for* `\xeCJK_FullRight_symbol:N`.)

\xeCJK_check_FullRight:

```
\cs_new_protected_nopar:Npn \xeCJK_check_FullRight:
  {
    \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \group_align_safe_begin:
    \tl_case:NoTF \l_peek_token
      { \l__xeCJK_no_break_cs_case_tl }
      { \group_align_safe_end: \xeCJK_no_break: }
      { \group_align_safe_end: }
    \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \xeCJK_class_group_end:
  }
\cs_generate_variant:Nn \tl_case:NnTF { No }
```

(*End definition for* `\xeCJK_check_FullRight:`.)

\xeCJK_check_FullRight_symbol:Nw

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

```
\cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
  {
    \tl_new:N #2
    \seq_new:N #3
    \keys_define:nn { xeCJK / options }
      {
        #1  .code:n =
          {
            \seq_set_split:Nnn #3 { } {##1}
            \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
          } ,
        #1+ .code:n =
          {
            \tl_map_inline:nn {##1}
              { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
            \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
          } ,
        #1- .code:n =
          {
            \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {####1} }
            \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
          }
      }
  }
\cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
  {
    \tl_clear:N #1
    \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
    #4
  }
```

(*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。

```
\xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
  \l__xeCJK_no_break_cs_case_tl \l__xeCJK_no_break_cs_seq { } { }
```

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

\xeCJKnobreak 为保险起见，我们在这里用了一个循环。

```
\NewDocumentCommand \xeCJKnobreak { }
  {
    \bool_set_true:N \l__xeCJK_tmp_bool
    \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
      {
        \bool_if:NTF \l__xeCJK_tmp_bool
          {
            \bool_set_false:N \l__xeCJK_tmp_bool
            \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
          }
          { \skip_add:Nn \l__xeCJK_tmp_skip \tex_lastskip:D }
        \tex_unskip:D
      }
    \xeCJK_no_break:
    \bool_if:NF \l__xeCJK_tmp_bool { \skip_horizontal:N \l__xeCJK_tmp_skip }
  }
```

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

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```
\keys_define:nn { xeCJK / options }
  {
    CheckSingle .choice: ,
    CheckSingle / true  .code:n =
      {
        \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
          {
            \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
            \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
          }
      } ,
    CheckSingle / false .code:n =
      {
        \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
          { \cs_set_eq:NN  \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
      } ,
    CheckSingle      .default:n = { true } ,
    CJKchecksingle      .meta:n = { CheckSingle = true }
  }
```

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

\xeCJK_check_single:Nw

```
\cs_new_protected_nopar:Npn \xeCJK_check_single:Nw #1
  {
    \peek_catcode:NTF \c_catcode_letter_token
      { \xeCJK_check_single:NNw #1 }
      {
        \group_align_safe_begin:
        \token_if_other:NTF \l_peek_token
          { \group_align_safe_end: \xeCJK_check_single:NNw #1 }
          {
            \group_align_safe_end:
            \bool_if:nTF
              {
                \str_if_eq_x_p:nn { \token_get_arg_spec:N \l_peek_token } { } &&
                \exp_args:No \tl_if_single_token_p:n \l_peek_token            &&
                ( \exp_after:wN \token_if_other_p:N  \l_peek_token ||
                  \exp_after:wN \token_if_letter_p:N \l_peek_token )
              }
              { \exp_after:wN \xeCJK_check_single:NNw \exp_after:wN #1 }
              { \__xeCJK_check_single_save:N #1 }
          }
      }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_check_single:NNw #1#2
  {
    \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
      {
        \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
          {
            \bool_if:NTF \l__xeCJK_reserve_space_bool
              { \__xeCJK_check_single_save:N #1 #2 ~ }
              { \__xeCJK_check_single_space:NN #1#2 }
          }
          { \__xeCJK_check_single_save:N #1 #2 }
      }
      {
        \group_align_safe_begin:
        \token_if_other:NTF \l_peek_token
          {
            \group_align_safe_end:
            \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
              { \__xeCJK_check_single_space:NN #1#2 }
              { \__xeCJK_check_single_save:N #1 #2 }
          }
          {
            \token_if_cs:NTF \l_peek_token
              {
                \group_align_safe_end:
                \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
                  { \xeCJK_check_single_cs:NNn #1#2 { ~ } }
                  { \xeCJK_check_single_cs:NNn #1#2 { } }
              }
              {
                \group_align_safe_end:
                \bool_if:nTF
                  {
                    \l__xeCJK_plain_equation_bool &&
                    \token_if_math_toggle_p:N \l_peek_token
                  }
                  {
                    \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
                      { \xeCJK_check_single_equation:NNnNw #1 #2 { ~ } }
                      { \xeCJK_check_single_equation:NNnNw #1 #2 { } }
                  }
                  {
                    \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
                      { \__xeCJK_check_single_save:N #1 #2 ~ }
                      { \__xeCJK_check_single_save:N #1 #2 }
                  }
              }
          }
      }
  }
```

(*End definition for* \xeCJK_check_single:NNw.)

__xeCJK_check_single_space:NN

```
\cs_new_protected_nopar:Npn \__xeCJK_check_single_space:NN #1#2
  {
    \xeCJK_if_CJK_class:NTF #2
      {
        \xeCJK_if_CJK_class:NTF \l_peek_token
          { \__xeCJK_check_single_save:N #1 #2 }
          { \__xeCJK_check_single_save:N #1 #2 ~ }
      }
      { \__xeCJK_check_single_save:N #1 #2 ~ }
  }
```

(*End definition for* __xeCJK_check_single_space:NN.)

\xeCJK_check_single_equation:NNnNw

```
\cs_new_protected_nopar:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
  {
    \peek_catcode:NTF \c_math_toggle_token
      {
        \xeCJK_no_break: \__xeCJK_check_single_save:N #1
        \xeCJK_make_node:n { CJK-nobreak } #2 #4
      }
      { \__xeCJK_check_single_save:N #1 #2#3#4 }
  }
```

(*End definition for* \xeCJK_check_single_equation:NNnNw.)

\xeCJK_check_single_cs:NNn

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

```
! Forbidden control sequence found while scanning use of \use_ii:nn.
<inserted text>
                \par
 l.10 \item
```

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

```
\cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
  {
    \group_align_safe_begin:
    \tl_case:NoF \l_peek_token
      { \l__xeCJK_check_single_cs_case_tl }
      { \group_align_safe_end: \use_iii:nnn }
      { \xeCJK_check_single_env:nnNn }
      {
        \xeCJK_no_break: \__xeCJK_check_single_save:N #1
        \xeCJK_make_node:n { CJK-nobreak } #2
      }
      { \__xeCJK_check_single_save:N #1 #2#3 }
  }
\tl_new:N \l__xeCJK_check_single_cs_case_tl
\cs_generate_variant:Nn \tl_case:NnF { No }
```

(*End definition for* \xeCJK_check_single_cs:NNn.)

xeCJK_check_single_env:nnNn

```
\cs_new_protected_nopar:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
  {
    \group_align_safe_begin:
    \str_case_x:noTF {#4}
      { \l__xeCJK_inline_env_case_tl }
      { \group_align_safe_end: #2 }
      { \group_align_safe_end: #1 }
    #3 {#4}
  }
\cs_generate_variant:Nn \str_case_x:nnTF { no }
```

(*End definition for* \xeCJK_check_single_env:nnNn.)

NewLineCS

```
\xeCJK_cs_case_keys_define:nNNnn { NewLineCS }
  \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
  { \group_align_safe_end: \use_ii:nnn }
  {
    \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
      \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
  }
```

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

EnvCS

```
\xeCJK_cs_case_keys_define:nNNnn { EnvCS }
  \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
  { \group_align_safe_end: \use:n }
```

```
  {
    \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
      \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
  }
```

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

InlineEnv

```
\keys_define:nn { xeCJK / options }
  {
    InlineEnv  .code:n =
      {
        \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
        \__xeCJK_update_inline_env_case_tl:
      } ,
    InlineEnv+      .code:n =
      {
        \clist_map_inline:nn {#1}
          {
            \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
              { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
          }
        \__xeCJK_update_inline_env_case_tl:
      } ,
    InlineEnv-      .code:n =
      {
        \clist_map_inline:nn {#1}
          { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
        \__xeCJK_update_inline_env_case_tl:
      }
  }
\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:

```
\cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
  {
    \tl_clear:N \l__xeCJK_inline_env_case_tl
    \seq_map_inline:Nn \l__xeCJK_inline_env_seq
      { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { } } }
  }
\tl_new:N \l__xeCJK_inline_env_case_tl
```

(*End definition for* `\__xeCJK_update_inline_env_case_tl:`.)

PlainEquation

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

```
\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。

```
\NewDocumentCommand \xeCJKDeclareSubCJKBlock
  { s > { \TrimSpaces } m > { \TrimSpaces } m }
  {
    \xeCJK_declare_sub_char_class:nxx { CJK } {#2} {#3}
    \IfBooleanT {#1} { \xeCJKResetPunctClass }
  }
\@onlypreamble \xeCJKDeclareSubCJKBlock
```

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

\xeCJKCancelSubCJKBlock
\xeCJKRestoreSubCJKBlock

取消和恢复对 CJK 子区的声明。

```
\bool_new:N \l__xeCJK_sub_cancel_bool
\NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
  {
    \bool_if:NF \l__xeCJK_sub_cancel_bool
      {
        \bool_set_true:N \l__xeCJK_sub_cancel_bool
        \__xeCJK_sub_restore_or_cancel:x {#2}
        \IfBooleanT {#1} { \xeCJKResetPunctClass }
      }
  }
\NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
  {
    \bool_if:NT \l__xeCJK_sub_cancel_bool
      {
        \bool_set_false:N \l__xeCJK_sub_cancel_bool
        \__xeCJK_sub_restore_or_cancel:x {#2}
        \IfBooleanT {#1} { \xeCJKResetPunctClass }
      }
  }
```

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

__xeCJK_sub_restore_or_cancel:n

```
\cs_new_protected_nopar:Npn \__xeCJK_sub_restore_or_cancel:n #1
  {
    \clist_map_inline:nn {#1}
      {
        \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
          {
            \xeCJK_declare_char_class:nx
              { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
              { \tl_use:c { g__xeCJK_CJK/##1_range_clist } }
          }
          { \__xeCJK_error:nx { SubBlock-undefined } {##1} }
      }
  }
\cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { x }
\__xeCJK_msg_new:nn { SubBlock-undefined }
  {
    The~CJK~sub~block~`#1'~is~undefined.\\\\
    Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
    to~declare~it.
  }
```

(*End definition for* `\__xeCJK_sub_restore_or_cancel:n`*.*)

\xeCJK_declare_sub_char_class:nnn

```
\cs_new_protected_nopar:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
  {
    \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
      {
        \xeCJK_new_class:n { #1/#2 }
        \__xeCJK_set_sub_class_toks:nn {#1} {#2}
        \xeCJK_new_sub_key:n {#2}
      }
    \xeCJK_declare_char_class:nn { #1/#2 } {#3}
  }
\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

```
\cs_new_protected_nopar:Npn \__xeCJK_set_sub_class_toks:nn #1#2
  {
    \seq_map_inline:Nn \g__xeCJK_base_class_seq
      {
        \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1}  {##1}
        \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
        \str_if_eq:nnTF {##1} { CJK }
```

```
          {
            \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
              { \__xeCJK_switch_font:nn {#1} {#2} }
          }
          {
            \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
              { \CJKsymbol }
              { \__xeCJK_switch_font:nn {#1} {#2} \CJKsymbol }
          }
      }
    \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
    \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
      {
        \xeCJK_copy_inter_class_toks:nnnn { #1/#2  } { #1/##1 } {#1} {#1}
        \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2  } {#1} {#1}
        \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
          { \__xeCJK_switch_font:nn {#2} {##1} }
        \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
          { \__xeCJK_switch_font:nn {##1} {#2} }
      }
    \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
    \__xeCJK_save_CJK_class:n { #1/#2 }
    \clist_map_inline:nn { CJK , FullLeft , FullRight }
      {
        \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
          { \__xeCJK_switch_font:nn {#2} {#1} }
      }
  }
```

(*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。

```
\cs_if_exist:NF \XeTeXglyphbounds
  {
    \__xeCJK_msg_new:nn { XeTeX-too-old }
      {
        \token_to_str:N \XeTeXglyphbounds \ is~not~defined.\\
        CJK~punctuation~kerning~will~not~be~available.\\\\
        You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
      }
    \__xeCJK_error:n { XeTeX-too-old }
    \AtEndOfPackage
      {
        \keys_define:nn { xeCJK / options }
          {
            PunctStyle / unknown .code:n =
              { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
          }
        \seq_gclear:N \g__xeCJK_punct_style_seq
        \xeCJKsetup { PunctStyle = plain }
      }
  }
```

`\xeCJKsetwidth` 手动设置参数中的标点符号的宽度。

```
\NewDocumentCommand \xeCJKsetwidth { s m m }
  {
    \IfBooleanTF {#1}
      {
        \tl_map_inline:xn {#2}
          { \tl_gset:cn { g__xeCJK_punct_bound_width/##1/tl } {#3} }
      }
      {
        \tl_map_inline:xn {#2}
          { \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#3} }
      }
```

```
  }
\@onlypreamble \xeCJKsetwidth
\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。

```
\NewDocumentCommand \xeCJKsetkern { m m m }
  { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
\@onlypreamble \xeCJKsetkern
```

(*End definition for* `\xeCJKsetkern`. *This function is documented on page* 10.)

`\c__xeCJK_left_tl`
`\c__xeCJK_right_tl`

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

(*End definition for* `\c__xeCJK_left_tl` *and* `\c__xeCJK_right_tl`.)

`AllowBreakBetweenPuncts`
`KaiMingPunct`
`LongPunct`
`MiddlePunct`
`PunctWidth`
`PunctBoundWidth`

相关选项声明。

```
\keys_define:nn { xeCJK / options }
  {
    AllowBreakBetweenPuncts .choice: ,
    AllowBreakBetweenPuncts / true  .code:n =
      {
        \bool_set_true:N \l__xeCJK_punct_breakable_bool
        \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
      } ,
    AllowBreakBetweenPuncts / false .code:n =
      {
        \bool_set_false:N \l__xeCJK_punct_breakable_bool
        \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_nobreak_kern:NN
      } ,
    AllowBreakBetweenPuncts      .default:n = { true } ,
    KaiMingPunct  .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
    KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
    KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
    LongPunct     .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
    LongPunct+    .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
    LongPunct-    .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
    MiddlePunct   .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
    MiddlePunct+  .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
    MiddlePunct-  .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
    PunctWidth      .tl_gset:N = \g__xeCJK_punct_width_tl ,
    PunctBoundWidth .tl_gset:N = \g__xeCJK_punct_bound_width_tl ,
    PunctWidth      .value_required: ,
    PunctBoundWidth .value_required:
  }
\bool_new:N \l__xeCJK_punct_breakable_bool
```

(*End definition for* `AllowBreakBetweenPuncts` *and others. These functions are documented on page* 5.)

相关选项定义的辅助函数。

```
\clist_new:N \g__xeCJK_special_punct_clist
\clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , middle }
\cs_new_nopar:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
\cs_new_nopar:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
\clist_map_inline:Nn \g__xeCJK_special_punct_clist
  { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
\cs_new_protected_nopar:Npn \__xeCJK_set_special_punct:nn #1#2
  {
    \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
      { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
    \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
    \tl_map_inline:xn {#2}
      {
        \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
        \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_add_special_punct:nn #1#2
  {
```

```
    \tl_map_inline:xn {#2}
      {
        \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
          {
            \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
            \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
          }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_sub_special_punct:nn #1#2
  {
    \tl_map_inline:xn {#2}
      {
        \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
        \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
      }
  }
```

判断一个标点符号是否为全角右标点和长标点符号。

```
\prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
  {
    \if_int_compare:w \xeCJK_token_value_class:N #1 = \xeCJK_class_num:n { FullRight }
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\clist_map_inline:Nn \g__xeCJK_special_punct_clist
  {
    \exp_args:Nc
    \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
      {
        \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
          \prg_return_true: \else: \prg_return_false: \fi:
      }
  }
```

一些用于记录的辅助函数。

```
\cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
  { c__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
\cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn #1#2#3
  { c__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/#3/tl }
\cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
  { \use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
\cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
  { \use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
\cs_new_protected_nopar:Npn \__xeCJK_save_punct_dim:nnn #1#2#3
  { \tl_const:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } { \dim_eval:n {#3} } }
\cs_new_protected_nopar:Npn \__xeCJK_save_punct_dim:nnnn #1#2#3#4
 { \tl_const:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } { \dim_eval:n {#4} } }
```

定义标点处理模板。

```
\DeclareObjectType { xeCJK / punctuation } { \c_zero }
\DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_zero }
  {
    enabled-global-setting  : boolean = true ,
    fixed-punct-width       : length  = \c_max_dim ,
    fixed-punct-ratio       : real    = \c_one_fp ,
    mixed-punct-width       : length  = \KeyValue { fixed-punct-width } ,
    mixed-punct-ratio       : real    = \KeyValue { fixed-punct-ratio } ,
    middle-punct-width      : length  = \KeyValue { fixed-punct-width } ,
    middle-punct-ratio      : real    = \KeyValue { fixed-punct-ratio } ,
    fixed-margin-width      : length  = \c_max_dim ,
    fixed-margin-ratio      : real    = \c_one_fp ,
    mixed-margin-width      : length  = \KeyValue { fixed-margin-width } ,
    mixed-margin-ratio      : real    = \KeyValue { fixed-margin-ratio } ,
    middle-margin-width     : length  = \KeyValue { fixed-margin-width } ,
    middle-margin-ratio     : real    = \KeyValue { fixed-margin-ratio } ,
    bound-punct-width       : length  = \c_max_dim ,
    bound-punct-ratio       : real    = \c_nan_fp ,
    bound-margin-width      : length  = \c_max_dim ,
    bound-margin-ratio      : real    = \c_zero_fp ,
```

```
    enabled-hanging         : boolean = false ,
    add-min-bound-to-margin : boolean = false ,
    optimize-margin         : boolean = false ,
    margin-minimum          : length  = \c_zero_dim ,
    enabled-kerning         : boolean = true ,
    min-bound-to-kerning    : boolean = false ,
    kerning-total-width     : length  = \c_max_dim ,
    kerning-total-ratio     : real    = 0.75 ,
    optimize-kerning        : boolean = false ,
    same-align-margin       : length  = \c_max_dim ,
    same-align-ratio        : real    = \c_nan_fp ,
    different-align-margin  : length  = \c_max_dim ,
    different-align-ratio   : real    = \c_nan_fp ,
    kerning-margin-width    : length  = \c_max_dim ,
    kerning-margin-ratio    : real    = \c_one_fp ,
    kerning-margin-minimum  : length  = \c_zero_dim
  }
\DeclareTemplateCode { xeCJK / punctuation } { basic } { \c_zero }
  {
    enabled-global-setting  = \l__xeCJK_enabled_global_setting_bool ,
    fixed-punct-width       = \l__xeCJK_fixed_punct_width_dim ,
    fixed-punct-ratio       = \l__xeCJK_fixed_punct_ratio_fp ,
    mixed-punct-width       = \l__xeCJK_mixed_punct_width_dim ,
    mixed-punct-ratio       = \l__xeCJK_mixed_punct_ratio_fp ,
    middle-punct-width      = \l__xeCJK_middle_punct_width_dim ,
    middle-punct-ratio      = \l__xeCJK_middle_punct_ratio_fp ,
    fixed-margin-width      = \l__xeCJK_fixed_margin_width_dim ,
    fixed-margin-ratio      = \l__xeCJK_fixed_margin_ratio_fp ,
    mixed-margin-width      = \l__xeCJK_mixed_margin_width_dim ,
    mixed-margin-ratio      = \l__xeCJK_mixed_margin_ratio_fp ,
    middle-margin-width     = \l__xeCJK_middle_margin_width_dim ,
    middle-margin-ratio     = \l__xeCJK_middle_margin_ratio_fp ,
    bound-punct-width       = \l__xeCJK_bound_punct_width_dim ,
    bound-punct-ratio       = \l__xeCJK_bound_punct_ratio_fp ,
    bound-margin-width      = \l__xeCJK_bound_margin_width_dim ,
    bound-margin-ratio      = \l__xeCJK_bound_margin_ratio_fp ,
    enabled-hanging         = \l__xeCJK_enabled_hanging_bool ,
    add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
    optimize-margin         = \l__xeCJK_optimize_margin_bool ,
    margin-minimum          = \l__xeCJK_margin_minimum_dim ,
    enabled-kerning         = \l__xeCJK_enabled_kerning_bool ,
    min-bound-to-kerning    = \l__xeCJK_min_bound_to_kerning_bool ,
    kerning-total-width     = \l__xeCJK_kerning_total_width_dim ,
    kerning-total-ratio     = \l__xeCJK_kerning_total_ratio_fp ,
    optimize-kerning        = \l__xeCJK_optimize_kerning_bool ,
    same-align-margin       = \l__xeCJK_same_align_margin_dim ,
    same-align-ratio        = \l__xeCJK_same_align_ratio_fp ,
    different-align-margin  = \l__xeCJK_different_align_margin_dim ,
    different-align-ratio   = \l__xeCJK_different_align_ratio_fp ,
    kerning-margin-width    = \l__xeCJK_kerning_margin_width_dim ,
    kerning-margin-ratio    = \l__xeCJK_kerning_margin_ratio_fp ,
    kerning-margin-minimum  = \l__xeCJK_kerning_margin_minimum_dim
  }
  { \AssignTemplateKeys }
```

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

```
\cs_new_protected_nopar:Npn \xeCJK_get_punct_bounds:NN #1#2
  {
    \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { glue } {#1} {#2} }
      {
        \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
          {
            \__xeCJK_save_punct_dim:nnnn { rule }   {#1} {#2} { \c_zero_dim }
            \__xeCJK_save_punct_dim:nnnn { glue }   {#1} {#2} { \c_zero_dim }
            \__xeCJK_save_punct_dim:nnnn { plus }   {#1} {#2} { \c_zero_dim }
            \__xeCJK_save_punct_dim:nnnn { minus }  {#1} {#2} { \c_zero_dim }
            \__xeCJK_save_punct_dim:nnnn { offset } {#1} {#2} { \c_zero_dim }
          \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_left_tl  {#2} { \c_zero_dim }
```

```
          \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
          }
          {
            { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
            \dim_set:Nn \l__xeCJK_bound_dim
              { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
            \dim_set:Nn \l__xeCJK_reverse_bound_dim
              {
                \__xeCJK_use_punct_dim:nnn { bound }
                  { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl }
                  {#2}
              }
            \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
            \xeCJK_punct_margin_process:NN {#1} {#2}
            \xeCJK_punct_offset_process:NN {#1} {#2}
          }
      }
  }
\dim_new:N \l__xeCJK_bound_dim
\dim_new:N \l__xeCJK_reverse_bound_dim
```

(*End definition for* *xeCJK_get_punct_bounds:NN*.)

\xeCJK_get_punct_kerning:NN 标点挤压。

```
\cs_new_protected_nopar:Npn \xeCJK_get_punct_kerning:NN #1#2
  {
    \tl_if_exist:cF { \__xeCJK_punct_dim_csname:nnn { kern } {#1} {#2} }
      {
        \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
          {
            \__xeCJK_save_punct_dim:nnnn { kern } {#1} {#2} { \c_zero_dim }
            \__xeCJK_save_punct_dim:nnnn { break_kern } {#1} {#2} { \c_zero_dim }
          }
          {
            \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
            \xeCJK_punct_kerning_process:NN {#1} {#2}
          }
      }
  }
\cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
```

(*End definition for* *xeCJK_get_punct_kerning:NN*.)

\xeCJK_punct_margin_process:NN

```
\cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
  {
    \dim_set:Nn \l__xeCJK_tmp_dim
      {
        \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
          {
            \cs_if_exist_use:cTF { g__xeCJK_punct_width/#2/tl }
              { \use_none:n }
              {
                \tl_if_empty:NTF \g__xeCJK_punct_width_tl
                  { \use:n }
                  { \g__xeCJK_punct_width_tl \use_none:n }
              }
          }
          { \use:n }
          {
            \__xeCJK_punct_if_middle:NTF {#2}
              { \__xeCJK_punct_width_or_ratio:nN { middle } {#2} }
              {
                \__xeCJK_punct_if_mixed_width:NTF {#2}
                  { \__xeCJK_punct_width_or_ratio:nN { mixed } {#2} }
                  { \__xeCJK_punct_width_or_ratio:nN { fixed } {#2} }
              }
          }
      }
    \dim_set:Nn \l__xeCJK_tmp_dim
```

```
      {
        \dim_max:nn
          { \l__xeCJK_margin_minimum_dim }
          {
            \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
              {
                \__xeCJK_punct_if_middle:NTF {#2}
                  {
                    (
                     \l__xeCJK_tmp_dim - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
                    ) / \c_two
                  }
                  {
                    \bool_if:NTF \l__xeCJK_optimize_margin_bool
                      {
                        \dim_max:nn
                       { \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
                      }
                      { \use:n }
                      {
                        \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim
                        - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
                      }
                  }
              }
              {
                \bool_if:NTF \l__xeCJK_optimize_margin_bool
                  { \dim_min:nn { \l__xeCJK_bound_dim } }
                  { \use:n }
                  {
                    \__xeCJK_punct_if_middle:NTF {#2}
                      {
                     \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
                          { \l__xeCJK_middle_margin_width_dim }
                          {
                            \fp_use:N \l__xeCJK_middle_margin_ratio_fp
                            \etex_dimexpr:D
                        ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / \c_two
                            \scan_stop:
                          }
                      }
                      {
                        \__xeCJK_punct_if_mixed_width:NTF {#2}
                          { \__xeCJK_margin_width_or_ratio:n { mixed } }
                          { \__xeCJK_margin_width_or_ratio:n { fixed } }
                      }
                  }
              }
          }
      }
    \__xeCJK_save_punct_dim:nnnn { glue } {#1} {#2} { \l__xeCJK_tmp_dim }
    \__xeCJK_save_punct_dim:nnnn { plus } {#1} {#2}
      {
        \dim_max:nn { \c_zero_dim }
          {
            \__xeCJK_punct_if_middle:NTF {#2}
              {
                ( \__xeCJK_use_punct_dim:nn { width } {#2} -
                  \__xeCJK_use_punct_dim:nn { dimen } {#2} ) / \c_two
                - \l__xeCJK_tmp_dim
              }
              { \l__xeCJK_bound_dim - \l__xeCJK_tmp_dim }
          }
      }
    \__xeCJK_save_punct_dim:nnnn { minus } {#1} {#2}
      {
        \dim_max:nn { \c_zero_dim }
          {
            \__xeCJK_punct_if_middle:NTF {#2}
```

```
            { .5 \l__xeCJK_tmp_dim }
            { \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim }
        }
    }
}
```

(*End definition for* \xeCJK_punct_margin_process:NN.)

\xeCJK_punct_offset_process:NN

```
\cs_new_protected_nopar:Npn \xeCJK_punct_offset_process:NN #1#2
  {
    \dim_set:Nn \l__xeCJK_tmp_dim
      {
        \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
          {
            \cs_if_exist_use:cTF { g__xeCJK_punct_bound_width/#2/tl }
              { \use_none:n }
              {
                \tl_if_empty:NTF \g__xeCJK_punct_bound_width_tl
                  { \use:n }
                  { \g__xeCJK_punct_bound_width_tl \use_none:n }
              }
          }
          { \use:n }
          { \__xeCJK_punct_width_or_ratio:nN { bound } {#2} }
      }
    \dim_set:Nn \l__xeCJK_tmp_dim
      {
        \bool_if:NTF \l__xeCJK_enabled_hanging_bool
          { \use:n }
          { \dim_max:nn { \l__xeCJK_margin_minimum_dim } }
          {
            \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
              {
                \__xeCJK_punct_if_middle:NTF {#2}
                  {
                    \l__xeCJK_tmp_dim
                    - ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} )
                    - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
                  }
                  {
                    \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim
                    - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
                  }
              }
              {
                \bool_if:NTF \l__xeCJK_optimize_margin_bool
                  { \dim_min:nn { \l__xeCJK_bound_dim } }
                  { \use:n }
                  { \__xeCJK_margin_width_or_ratio:n { bound } }
              }
          }
      }
    \__xeCJK_save_punct_dim:nnnn { offset } {#1} {#2} { \l__xeCJK_tmp_dim }
    \__xeCJK_save_punct_dim:nnnn { rule } {#1} {#2}
      { - \l__xeCJK_bound_dim + \l__xeCJK_tmp_dim }
  }
```

(*End definition for* \xeCJK_punct_offset_process:NN.)

__xeCJK_punct_width_or_ratio:nN

```
\cs_new_nopar:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
  {
    \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
      { \use:c { l__xeCJK_#1_punct_width_dim } }
      {
        \fp_compare:nNnTF { \use:c { l__xeCJK_#1_punct_ratio_fp } } ? \c_zero_fp
          { \c_max_dim }
          {
            \fp_use:c { l__xeCJK_#1_punct_ratio_fp }
```

```
            \etex_dimexpr:D \__xeCJK_use_punct_dim:nn { width } {#2} \scan_stop:
          }
      }
  }
```

(*End definition for* `\__xeCJK_punct_width_or_ratio:nN`.)

`\__xeCJK_margin_width_or_ratio:n`

```
\cs_new_nopar:Npn \__xeCJK_margin_width_or_ratio:n #1
  {
    \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
      { \use:c { l__xeCJK_#1_margin_width_dim } }
      {
        \fp_use:c { l__xeCJK_#1_margin_ratio_fp }
        \etex_dimexpr:D \l__xeCJK_bound_dim \scan_stop:
      }
    \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
      { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
  }
```

(*End definition for* `\__xeCJK_margin_width_or_ratio:n`.)

`\xeCJK_punct_kerning_process:NN`

```
\cs_new_protected_nopar:Npn \xeCJK_punct_kerning_process:NN #1#2
  {
    \dim_set:Nn \l__xeCJK_tmp_dim
      {
        \bool_if:nTF
          {
            \l__xeCJK_enabled_global_setting_bool &&
            \tl_if_exist_p:c { g__xeCJK_punct/kern/#1/#2/tl }
          }
          { \tl_use:c { g__xeCJK_punct/kern/#1/#2/tl } }
          {
            \bool_if:NTF \l__xeCJK_enabled_kerning_bool
              { \__xeCJK_calc_kerning_margin:NN {#1} {#2} }
              { \__xeCJK_original_kerning_margin:NN {#1} {#2} }
          }
      }
    \__xeCJK_save_punct_dim:nnnn { kern } {#1} {#2}
      {
        \l__xeCJK_tmp_dim
        - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
        - ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl  {#2} )
      }
    \__xeCJK_punct_if_right:NF {#2}
      {
        \__xeCJK_punct_if_right:NT {#1}
          {
            \__xeCJK_save_punct_dim:nnnn { break_kern } {#1} {#2}
              {
                \l__xeCJK_tmp_dim
                - ( \__xeCJK_use_punct_dim:nnn { offset } \c__xeCJK_right_tl {#1} )
                - ( \__xeCJK_use_punct_dim:nnn { offset } \c__xeCJK_left_tl  {#2} )
              }
          }
      }
  }
```

(*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。

```
\cs_new_nopar:Npn \__xeCJK_original_kerning_margin:NN #1#2
  {
    \dim_eval:n
      {
        \__xeCJK_use_punct_dim:nnn
          { \__xeCJK_punct_if_right:NTF {#1} { glue } { bound } }
          { \c__xeCJK_right_tl } {#1} +
        \__xeCJK_use_punct_dim:nnn
```

```
          { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
          { \c__xeCJK_left_tl  } {#2}
      }
  }
```

(*End definition for* __xeCJK_original_kerning_margin:NN.)

__xeCJK_calc_kerning_margin:NN

```
\cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:NN #1#2
  {
    \dim_max:nn
      { \l__xeCJK_kerning_margin_minimum_dim }
      {
        \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
          { \__xeCJK_punct_min_bound:NN  {#1} {#2} }
          {
            \bool_if:NTF \l__xeCJK_optimize_kerning_bool
              { \dim_max:nn { \__xeCJK_punct_min_bound:NN  {#1} {#2} } }
              { \use:n }
              {
                \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
               { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
                  {
                    \fp_compare:nNnTF \l__xeCJK_kerning_total_ratio_fp ? \c_zero_fp
                      {
                        \xeCJK_if_same_class:NNTF {#1} {#2}
                          { \__xeCJK_kerning_width_or_ratio:nNN { same } }
                          { \__xeCJK_kerning_width_or_ratio:nNN { different } }
                      }
                      {
                        \__xeCJK_calc_kerning_margin:nNN
                          {
                            \fp_use:N \l__xeCJK_kerning_total_ratio_fp
                            \etex_dimexpr:D
                              \__xeCJK_use_punct_dim:nn { width } {#1} +
                              \__xeCJK_use_punct_dim:nn { width } {#2}
                            \scan_stop:
                          }
                      }
                  }
                  {#1} {#2}
              }
          }
      }
  }
```

(*End definition for* __xeCJK_calc_kerning_margin:NN.)

__xeCJK_kerning_width_or_ratio:nNN

```
\cs_new_nopar:Npn \__xeCJK_kerning_width_or_ratio:nNN #1#2#3
  {
    \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
      { \use:c { l__xeCJK_#1_align_margin_dim } }
      {
        \fp_compare:nNnTF { \use:c { l__xeCJK_#1_align_ratio_fp } } ? \c_zero_fp
          {
            \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
              { \l__xeCJK_kerning_margin_width_dim \use_none:n }
              { \fp_use:N \l__xeCJK_kerning_margin_ratio_fp \use:n }
          }
          { \fp_use:c { l__xeCJK_#1_align_ratio_fp } \use:n }
         { \etex_dimexpr:D \__xeCJK_original_kerning_margin:NN {#2} {#3} \scan_stop: }
      }
  }
```

(*End definition for* __xeCJK_kerning_width_or_ratio:nNN.)

__xeCJK_punct_min_bound:NN

```
\cs_new_nopar:Npn \__xeCJK_punct_min_bound:NN #1#2
  {
```

```
    \dim_max:nn
      {
        \dim_min:nn
          { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl  {#1} }
          { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} }
      }
      {
        \dim_min:nn
          { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl  {#2} }
          { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#2} }
      }
  }
```

(*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。

```
\cs_new_nopar:Npn \__xeCJK_calc_kerning_margin:nNN #1#2#3
  {
    \dim_eval:n
      {
        (#1)
        - ( \__xeCJK_use_punct_dim:nnn
              { \__xeCJK_punct_if_right:NTF {#2} { bound } { glue } }
              { \c__xeCJK_left_tl } {#2} )
        - ( \__xeCJK_use_punct_dim:nnn
              { \__xeCJK_punct_if_right:NTF {#3} { glue } { bound } }
              { \c__xeCJK_right_tl } {#3} )
        - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
        - ( \__xeCJK_use_punct_dim:nn { dimen } {#3} )
      }
  }
```

(*End definition for* __xeCJK_calc_kerning_margin:nNN.)

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

```
\cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
  {
    \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_left_tl {#1}
      { \xeCJK_glyph_bounds:NN \c_one {#1} }
    \__xeCJK_save_punct_dim:nnnn { bound } \c__xeCJK_right_tl {#1}
      { \xeCJK_glyph_bounds:NN \c_three {#1} }
    \dim_set:Nn \l__xeCJK_tmp_dim
      {
        ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl  {#1} ) +
        ( \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
      }
    \__xeCJK_save_punct_dim:nnn { width } {#1}
      { \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
    \__xeCJK_save_punct_dim:nnn { dimen } {#1}
      { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim }
    \__xeCJK_punct_if_long:NT {#1}
      {
        \__xeCJK_save_punct_dim:nnnn { kern } {#1} {#1}
          {
            \str_case:nnTF {#1}
              { { ^^^^2025 } { } { ^^^^2026 } { } }
              { \c_zero_dim }
              { - \l__xeCJK_tmp_dim }
          }
      }
  }
\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。

```
\cs_new_nopar:Npn \xeCJK_glyph_bounds:NN #1#2
  {
    \dim_use:N \XeTeXglyphbounds #1 ~
```

```
    \XeTeXcharglyph \xeCJK_token_value_charcode:N #2 \exp_stop_f:
  }
```

(*End definition for* `\xeCJK_glyph_bounds:NN`.)

PunctStyle

```
\keys_define:nn { xeCJK / options }
  {
    PunctStyle .choice: ,
    PunctStyle                .default:n = { quanjiao } ,
    PunctStyle / halfwidth      .meta:n = { PunctStyle = banjiao } ,
    PunctStyle / fullwidth      .meta:n = { PunctStyle = quanjiao } ,
    PunctStyle / mixedwidth     .meta:n = { PunctStyle = kaiming } ,
    PunctStyle / marginkerning .meta:n = { PunctStyle = hangmobanjiao } ,
    PunctStyle / plain          .code:n =
      { \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl } ,
    PunctStyle / unknown        .code:n =
      {
        \IfInstanceExistTF { xeCJK / punctuation } { \l_keys_value_tl }
          { \tl_set:Nx \l_xeCJK_punct_style_tl { \l_keys_value_tl } }
          { \__xeCJK_error:nx { punct-style-unknown } { \l_keys_value_tl } }
      }
  }
\tl_new:N \l_xeCJK_punct_style_tl
\tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
\__xeCJK_msg_new:nn { punct-style-unknown }
  {
    Punctuation~style~"#1"~is~unknown. \\\\
    The~available~styles~are~listed~as~follow.\\\\
    "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq { ~and~ } { ,~ } { ,~and~ }".\\
  }
```

(*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。

```
\NewDocumentCommand \xeCJKDeclarePunctStyle { > { \TrimSpaces } m m }
  {
    \IfInstanceExistTF { xeCJK / punctuation } {#1}
      { \__xeCJK_warning:nx { punct-style-already-defined } {#1} }
      { \seq_gput_right:Nx \g__xeCJK_punct_style_seq {#1} }
    \exp_args:Nnx \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
  }
\seq_new:N \g__xeCJK_punct_style_seq
\__xeCJK_msg_new:nn { punct-style-already-defined }
  {
    Punctuation~style~"#1"~is~already~defined!. \\\\
    The~existing~style~of~"#1"~will~be~overwritten.\\
  }
\@onlypreamble \xeCJKDeclarePunctStyle
```

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

\xeCJKEditPunctStyle 对已有的标点处理风格进行修改。

```
\NewDocumentCommand \xeCJKEditPunctStyle { > { \TrimSpaces } m m }
  {
    \IfInstanceExistTF { xeCJK / punctuation } {#1}
      { \exp_args:Nnx \EditInstance { xeCJK / punctuation } {#1} {#2} }
      { \__xeCJK_error:nx { punct-style-unknown } {#1} }
  }
\@onlypreamble \xeCJKEditPunctStyle
```

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

默认设置即为全角格式。

```
\xeCJKDeclarePunctStyle { quanjiao } { }
```

```
\xeCJKDeclarePunctStyle { hangmobanjiao } { enabled-kerning = false }
```

```
\xeCJKDeclarePunctStyle { banjiao }
  {
    fixed-punct-ratio   = 0.5  ,
    optimize-margin     = true ,
    kerning-total-ratio = 0.5  ,
```

```
    optimize-kerning    = true
  }
\xeCJKDeclarePunctStyle { kaiming }
  {
    fixed-punct-ratio   = 0.5  ,
    mixed-punct-ratio   = 0.8  ,
    optimize-margin     = true ,
    kerning-total-ratio = 0.5  ,
    optimize-kerning    = true
  }
\xeCJKDeclarePunctStyle { CCT }
  {
    fixed-punct-ratio   = 0.7  ,
    optimize-margin     = true ,
    kerning-total-ratio = 0.6  ,
    optimize-kerning    = true
  }
```

5.11 后备字体

AutoFallBack 后备字体的宏包选项声明。

```
\keys_define:nn { xeCJK / options }
  {
    AutoFallBack .choice: ,
    AutoFallBack / true  .code:n =
      {
        \cs_if_eq:NNF \CJKsymbol \xeCJK_fallback_test_glyph:N
          {
            \cs_set_eq:NN \__xeCJK_fallback_save_CJKsymbol:N \CJKsymbol
            \cs_set_eq:NN \CJKsymbol \xeCJK_fallback_test_glyph:N
          }
      } ,
    AutoFallBack / false .code:n =
      {
        \cs_if_eq:NNT \CJKsymbol \xeCJK_fallback_test_glyph:N
          { \cs_set_eq:NN \CJKsymbol \__xeCJK_fallback_save_CJKsymbol:N }
      } ,
    AutoFallBack      .default:n = { true } ,
    fallback             .meta:n = { AutoFallBack = true }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_fallback_test_glyph:N #1
  {
    \xeCJK_glyph_if_exist:NTF {#1}
      { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
      {
        \xeCJK_class_group_begin:
        \tl_set_eq:NN \l__xeCJK_fallback_family_tl \l_xeCJK_family_tl
        \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_fallback_family_tl/FallBack }
        \xeCJK_class_group_end:
      }
  }
```

(*End definition for* `\xeCJK_fallback_test_glyph:N`*.*)

\xeCJK_fallback_loop:Nn 循环测试后备字体是否包含字符 #1。若后备字体中存在该字符或者再没有后备字体，则结束循环。当前字体族没有备用字体时，使用 \CJKfamilydefault 的设置。

```
\cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
  {
    \xeCJK_family_if_exist:xTF {#2}
      {
        \xeCJK_select_font:x {#2}
        \tl_set:Nx \l__xeCJK_fallback_family_tl {#2}
        \xeCJK_glyph_if_exist:NTF {#1}
          { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
```

```
          { \xeCJK_fallback_loop:Nn {#1} { \l__xeCJK_fallback_family_tl/FallBack } }
      }
      {
        \str_if_eq_x:nnTF { \CJKfamilydefault } { \l_xeCJK_family_tl }
          {
            \__xeCJK_warning:nxxx { missing-glyph }
              { \l__xeCJK_fallback_family_tl } {#1}
              { \int_to_hexadecimal:n { `#1 } }
            \__xeCJK_fallback_save_CJKsymbol:N {#1}
          }
          {
            \tl_set:Nx \l_xeCJK_family_tl { \CJKfamilydefault }
            \xeCJK_fallback_loop:Nn {#1} { \l_xeCJK_family_tl }
          }
      }
  }
\__xeCJK_msg_new:nn { missing-glyph }
  {
    CJKfamily~`\__xeCJK_msg_family_map:n {#1}'~
    ( \prop_get:Nn \g__xeCJK_family_font_name_prop {#1} )~
    does~not~contain~glyph~`#2'~(U+#3).\\
  }
```

(*End definition for* `\xeCJK_fallback_loop:Nn`.)

`\setCJKfallbackfamilyfont`

```
\NewDocumentCommand \setCJKfallbackfamilyfont { m O { } m }
  { \use:x { \xeCJK_set_family_fallback:nnn {#1} {#2} {#3} } }
```

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

`\xeCJK_set_family_fallback:nnn`

```
\cs_new_protected_nopar:Npn \xeCJK_set_family_fallback:nnn #1#2#3
  {
    \group_begin:
    \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
    \prop_get:NVNF \g__xeCJK_family_font_name_prop
      \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
      { \tl_clear:N \l__xeCJK_font_name_tl }
    \clist_map_inline:nn {#3}
      {
        \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
        \__xeCJK_get_sub_features:Vn \l__xeCJK_fallback_family_tl {##1}
        \clist_put_left:Nn \l__xeCJK_sub_font_options_clist {#2}
        \xeCJK_set_family:VVV \l__xeCJK_fallback_family_tl
          \l__xeCJK_sub_font_options_clist \l__xeCJK_sub_font_name_tl
      }
    \group_end:
  }
\tl_new:N \l__xeCJK_fallback_family_tl
```

(*End definition for* `\xeCJK_set_family_fallback:nnn`.)

5.12 CJK 字体族声明方式

```
\bool_new:N \g__xeCJK_auto_fake_bold_bool
\bool_new:N \g__xeCJK_auto_fake_slant_bool
\fp_new:N \g__xeCJK_embolden_factor_fp
\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。

```
\keys_define:nn { xeCJK / options }
  {
    AutoFakeBold .choices:nn = { true , false }
      { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_bold_bool } ,
    AutoFakeBold / unknown .code:n =
      {
        \bool_gset_true:N  \g__xeCJK_auto_fake_bold_bool
        \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
      } ,
```

```
    AutoFakeBold  .default:n  = { true } ,
    AutoFakeSlant .choices:nn = { true , false }
      { \use:c { bool_gset_ \l_keys_choice_tl :N } \g__xeCJK_auto_fake_slant_bool } ,
    AutoFakeSlant / unknown  .code:n =
      {
        \bool_gset_true:N  \g__xeCJK_auto_fake_slant_bool
        \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
      } ,
    AutoFakeSlant  .default:n = { true } ,
    EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
    SlantFactor    .fp_gset:N = \g__xeCJK_slant_factor_fp ,
    BoldFont  .meta:n = { AutoFakeBold  = true } ,
    boldfont  .meta:n = { AutoFakeBold  = true } ,
    SlantFont .meta:n = { AutoFakeSlant = true } ,
    slantfont .meta:n = { AutoFakeSlant = true }
  }
```

(*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。

```
\seq_new:N \g__xeCJK_sub_key_seq
\cs_new_protected_nopar:Npn \xeCJK_new_sub_key:n #1
  {
    \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
    \keys_define:nn { xeCJK / features }
      {
        #1 .code:n =
          {
            \tl_if_blank:nTF {##1}
              {
                \prop_clear:N \l__xeCJK_sub_key_prop
                \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
                \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
              }
              {
                \str_if_eq:nnTF {##1} { * }
                  { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
                  { \__xeCJK_get_sub_features:nn {#1} {##1} }
              }
          } ,
        #1 .default:n = { }
      }
  }
```

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

`__xeCJK_get_sub_features:nn`
`\__xeCJK_get_sub_features:w`

```
\cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
  {
    \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
    \clist_clear:N \l__xeCJK_sub_font_options_clist
    \exp_after:wN \__xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
      \q_mark [ \q_nil ] \q_mark \q_stop
    \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
      { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
      { \tl_replace_all:NnV \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
    \prop_put:Nnx \l__xeCJK_sub_key_prop {#1}
      {
        { \exp_not:V \l__xeCJK_sub_font_options_clist }
        { \exp_not:V \l__xeCJK_sub_font_name_tl }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
  {
    \quark_if_nil:nTF {#2}
      { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
      {
        \tl_set:Nx \l__xeCJK_sub_font_name_tl
          { \xeCJK_tl_remove_outer_braces:n {#3} }
        \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
```

```
          { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
          { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
      }
  }
\tl_new:N \l__xeCJK_sub_family_name_tl
\tl_new:N \l__xeCJK_sub_font_name_tl
\clist_new:N \l__xeCJK_sub_font_options_clist
\cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
\cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
```

(*End definition for* `\__xeCJK_get_sub_features:nn` *and* `\__xeCJK_get_sub_features:w`.)

FallBack

```
\xeCJK_new_sub_key:n { FallBack }
```

(*End definition for* `FallBack`. *This function is documented on page 7.*)

BoldFont
ItalicFont

调用字体的属性声明，同 fontspec 宏包。

```
\keys_define:nn { xeCJK / features }
  {
    BoldFont   .tl_set:N = \l__xeCJK_font_name_bf_tl ,
    ItalicFont .tl_set:N = \l__xeCJK_font_name_it_tl
  }
```

(*End definition for* `BoldFont` *and* `ItalicFont`.)

AutoFakeBold
AutoFakeSlant

```
\keys_define:nn { xeCJK / features }
  {
    AutoFakeBold  .choice: ,
    AutoFakeBold / true    .code:n =
      {
        \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
        \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
      } ,
    AutoFakeBold / false   .code:n =
      { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
    AutoFakeBold / unknown .code:n =
      {
        \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
        \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
      } ,
    AutoFakeBold .default:n  = { true } ,
    AutoFakeSlant  .choice: ,
    AutoFakeSlant / true    .code:n =
      {
        \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
        \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
      } ,
    AutoFakeSlant / false   .code:n =
      { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
    AutoFakeSlant / unknown .code:n =
      {
        \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
        \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
      } ,
    AutoFakeSlant .default:n  = { true }
  }
```

(*End definition for* `AutoFakeBold` *and* `AutoFakeSlant`.)

`__xeCJK_set_family_initial:`

```
\cs_new_protected_nopar:Npn \__xeCJK_set_family_initial:
  {
    \int_gincr:N \g__xeCJK_family_int
    \prop_clear:N \l__xeCJK_sub_key_prop
    \tl_clear:N \l__xeCJK_font_name_bf_tl
    \tl_clear:N \l__xeCJK_font_name_it_tl
    \clist_clear:N \l__xeCJK_fontspec_options_clist
    \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool  \g__xeCJK_auto_fake_bold_bool
    \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
```

```
    \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
    \fp_set_eq:NN \l__xeCJK_slant_factor_fp    \g__xeCJK_slant_factor_fp
  }
\int_new:N \g__xeCJK_family_int
\prop_new:N \l__xeCJK_sub_key_prop
\clist_new:N \l__xeCJK_fontspec_options_clist
\bool_new:N \l__xeCJK_auto_fake_bold_bool
\bool_new:N \l__xeCJK_auto_fake_slant_bool
\fp_new:N \l__xeCJK_embolden_factor_fp
\fp_new:N \l__xeCJK_slant_factor_fp
```

(*End definition for* __xeCJK_set_family_initial:.)

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

```
\cs_new_protected_nopar:Npn \xeCJK_set_family:nnn #1#2#3
  {
    \group_begin:
    \__xeCJK_set_family_initial:
    \tl_set:Nn \l__xeCJK_family_name_tl {#1}
    \clist_set:Nn \l__xeCJK_font_options_clist {#2}
    \tl_set:Nn \l__xeCJK_font_name_tl {#3}
    \clist_concat:NNN \l__xeCJK_font_options_clist
      \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
    \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_options_clist
    \keys_set_known:nVN { xeCJK / features }
      \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
    \__xeCJK_parse_font_shape:
    \__xeCJK_check_family:V \l__xeCJK_family_name_tl
    \__xeCJK_gset_family_cs:x { \l__xeCJK_family_name_tl }
    \__xeCJK_save_family_info:
    \__xeCJK_set_sub_block_family:
    \group_end:
  }
\tl_new:N \l__xeCJK_family_name_tl
\tl_new:N \l__xeCJK_font_name_tl
\clist_new:N \l__xeCJK_font_options_clist
\cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , VVV , Voo }
\cs_new_protected_nopar:Npn \xeCJK_set_family:xxx #1#2#3
  { \use:x { \xeCJK_set_family:nnn {#1} {#2} {#3} } }
```

(*End definition for* \xeCJK_set_family:nnn.)

__xeCJK_remove_duplicate_keys:N

```
\cs_new_protected_nopar:Npn \__xeCJK_remove_duplicate_keys:N #1
  {
    \prop_clear:N \l__xeCJK_font_options_prop
    \keyval_parse:NNV \__xeCJK_prop_put_aux:n \__xeCJK_prop_put_aux:nn #1
    \clist_clear:N #1
    \prop_map_inline:Nn \l__xeCJK_font_options_prop
      {
        \tl_set:No \l__xeCJK_tmp_tl { \use_ii:nn ##2 }
        \tl_if_blank:VTF \l__xeCJK_tmp_tl
          { \clist_put_right:No #1 { \use_i:nn ##2 } }
          {
            \clist_put_right:Nx #1
              { \exp_not:o { \use_i:nn ##2 } = { \exp_not:V \l__xeCJK_tmp_tl } }
          }
      }
  }
\prop_new:N \l__xeCJK_font_options_prop
\cs_generate_variant:Nn \keyval_parse:NNn { NNV }
\cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
  { \prop_put:Nnn \l__xeCJK_font_options_prop {#1} { {#1} {  } } }
\cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
  { \prop_put:Nnn \l__xeCJK_font_options_prop {#1} { {#1} {#2} } }
```

(*End definition for* __xeCJK_remove_duplicate_keys:N.)

__xeCJK_gset_family_cs:x

```
\cs_new_protected_nopar:Npn \__xeCJK_gset_family_cs:x #1
```

```
  {
    \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n {#1} }
      {
        \group_begin:
        \exp_not:n { \cs_set_eq:NN \__xeCJK_update_family:nn \use_none:nn }
        \exp_not:n { \fontspec_set_family:Nnn \l__xeCJK_fontspec_family_tl }
          { \exp_not:V \l__xeCJK_fontspec_options_clist }
          { \exp_not:V \l__xeCJK_font_name_tl }
      \__xeCJK_gset_family_nfss_cs:xx {#1} { \exp_not:N \l__xeCJK_fontspec_family_tl }
        \group_end:
      }
  }
\tl_new:N \l__xeCJK_fontspec_family_tl
```

(*End definition for* __xeCJK_gset_family_cs:x.)

__xeCJK_check_family:n

```
\cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
  {
    \prop_gpop:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
      {
        \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
          { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
        \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
      }
  }
\cs_generate_variant:Nn \__xeCJK_check_family:n { V }
\__xeCJK_msg_new:nn { CJKfamily-redef }
  { Redefining~CJKfamily~`\__xeCJK_msg_family_map:n {#1}'~(#2). }
```

(*End definition for* __xeCJK_check_family:n.)

__xeCJK_parse_font_shape:

```
\cs_new_protected_nopar:Npn \__xeCJK_parse_font_shape:
  {
    \tl_if_blank:VTF \l__xeCJK_font_name_bf_tl
      {
        \bool_if:NT \l__xeCJK_auto_fake_bold_bool
          {
            \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
              { AutoFakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
          }
      }
      {
        \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
          { BoldFont = { \exp_not:V \l__xeCJK_font_name_bf_tl } }
      }
    \tl_if_blank:VTF \l__xeCJK_font_name_it_tl
      {
        \bool_if:NT \l__xeCJK_auto_fake_slant_bool
          {
            \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
              { AutoFakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
          }
      }
      {
        \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
          { ItalicFont = { \exp_not:V \l__xeCJK_font_name_it_tl } }
      }
  }
```

(*End definition for* __xeCJK_parse_font_shape:.)

\g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop

```
\prop_new:N \g__xeCJK_family_name_prop
\prop_new:N \g__xeCJK_family_font_name_prop
\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_pro

__xeCJK_save_family_info:

```
\cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
  {
    \prop_gput:NVV \g__xeCJK_family_font_name_prop
      \l__xeCJK_family_name_tl \l__xeCJK_font_name_tl
    \prop_gput:NVV \g__xeCJK_family_font_options_prop
      \l__xeCJK_family_name_tl \l__xeCJK_font_options_clist
  }
```

(*End definition for* __xeCJK_save_family_info:.)

__xeCJK_set_sub_block_family:

```
\cs_new_protected_nopar:Npn \__xeCJK_set_sub_block_family:
  {
    \prop_map_inline:Nn \l__xeCJK_sub_key_prop
      {
        \tl_set:Nx \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/##1 }
        \quark_if_no_value:nTF {##2}
          { \__xeCJK_copy_sub_family:n {##1} }
          {
            \xeCJK_set_family:Voo \l__xeCJK_sub_family_name_tl
              { \use_i:nn ##2 } { \use_ii:nn ##2 }
          }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
  {
    \__xeCJK_check_family:V \l__xeCJK_sub_family_name_tl
    \prop_get:NVNT \g__xeCJK_family_font_name_prop
      \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
      {
        \prop_gput:NVV \g__xeCJK_family_font_name_prop
          \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
      }
    \prop_get:NVNT \g__xeCJK_family_font_options_prop
      \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
      {
        \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
        \prop_gput:NVV \g__xeCJK_family_font_options_prop
          \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist
      }
    \cs_gset_protected_nopar:cpx
      { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
      {
        \xeCJK_family_if_exist:xT { \l__xeCJK_family_name_tl }
          {
            \prop_get:NnNT \exp_not:N \g__xeCJK_family_name_prop
              { \l__xeCJK_family_name_tl } \exp_not:N \l__xeCJK_fontspec_family_tl
              {
                \__xeCJK_gset_family_nfss_cs:xx
                  { \l__xeCJK_sub_family_name_tl }
                  { \exp_not:N \l__xeCJK_fontspec_family_tl }
              }
          }
      }
  }
```

(*End definition for* __xeCJK_set_sub_block_family:.)

__xeCJK_copy_family:nn

```
\cs_new_protected_nopar:Npn \__xeCJK_copy_family:nn #1#2
  {
    \xeCJK_family_if_exist:xT {#2}
      {
        \tl_map_inline:nn
          {
            \g__xeCJK_family_name_prop
            \g__xeCJK_family_font_name_prop
            \g__xeCJK_family_font_options_prop
          }
```

```
          {
            \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
              { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmp_tl }
          }
        \cs_gset_eq:cc
          { \__xeCJK_family_nfss_csname:n {#1} }
          { \__xeCJK_family_nfss_csname:n {#2} }
      }
  }
\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` `\xeCJK_select_font:` `\xeCJK_select_font:x`

缓存当前字体的原始格式,以加速编译。

```
\tl_new:N \l_xeCJK_current_font_tl
\tl_set:Nn \l_xeCJK_current_font_tl { \__xeCJK_font_csname:n { \l_xeCJK_family_tl } }
\cs_new_nopar:Npn \__xeCJK_font_csname:n #1 { xeCJK/#1/\f@series/\f@shape/\f@size }
\cs_new_protected_nopar:Npn \xeCJK_select_font:
  {
    \cs_if_exist_use:cF { \l_xeCJK_current_font_tl }
      {
        \tl_set:Nx \l__xeCJK_current_coor_tl { \l_xeCJK_current_font_tl }
        \__xeCJK_family_use:x { \l_xeCJK_family_tl }
        \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
      }
  }
\cs_new_protected_nopar:Npn \xeCJK_select_font:x #1
  {
    \cs_if_exist_use:cF { \__xeCJK_font_csname:n {#1} }
      {
        \tl_set:Nx \l__xeCJK_current_coor_tl { \__xeCJK_font_csname:n {#1} }
        \__xeCJK_family_use:x {#1}
        \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
      }
  }
\tl_new:N \l__xeCJK_current_coor_tl
\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。

```
\cs_new_protected_nopar:Npn \__xeCJK_switch_font:nn #1#2
  {
    \str_if_eq:nnF {#1} {#2}
      {
        \__xeCJK_info:nxx { CJK-block } {#1} {#2}
        \str_if_eq:nnTF {#2} { CJK }
          { \xeCJK_select_font: }
          { \__xeCJK_block_select_font:n {#2} }
      }
  }
\__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。

```
\cs_new_protected_nopar:Npn \__xeCJK_block_select_font:n #1
  {
    \cs_if_exist_use:cF { \__xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
      {
        \tl_set:Nx \l__xeCJK_current_coor_tl
          { \__xeCJK_font_csname:n { \l_xeCJK_family_tl/#1 } }
        \xeCJK_family_if_exist:xF { \l_xeCJK_family_tl/#1 }
          {
            \__xeCJK_copy_family:xx { \l_xeCJK_family_tl/#1 }
              {
                \cs_if_exist:cTF
```

```
                    { \__xeCJK_family_csname:n { \CJKfamilydefault/#1 } }
                    { \CJKfamilydefault/#1 } { \l_xeCJK_family_tl }
                }
            }
          \__xeCJK_family_use:x { \l_xeCJK_family_tl/#1 }
          \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
        }
    }
```

(*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`

```
\cs_new_nopar:Npn \__xeCJK_family_csname:n #1 { xeCJK/family/#1 }
\cs_new_nopar:Npn \__xeCJK_family_nfss_csname:n #1 { xeCJK/family/nfss/#1 }
\cs_new_nopar:Npn \__xeCJK_family_use:x #1 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
\cs_new_protected_nopar:Npn \__xeCJK_gset_family_nfss_cs:xx #1#2
  {
    \prop_gput:Nxx \g__xeCJK_family_name_prop {#1} {#2}
    \cs_gset_protected_nopar:cpx { \__xeCJK_family_nfss_csname:n {#1} }
      {
        \exp_not:N \fontencoding { \c__xeCJK_encoding_tl }
        \tl_set:Nx \exp_not:N \f@family {#2}
        \exp_not:N \selectfont
      }
  }
\cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
```

(*End definition for* `\__xeCJK_family_csname:n` *and others.*)

`\xeCJK_family_if_exist:xTF`

```
\prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:x #1 { T , F , TF }
  {
    \cs_if_exist:cTF { \__xeCJK_family_nfss_csname:n {#1} }
      { \use_i:nn }
      { \cs_if_exist_use:cTF { \__xeCJK_family_csname:n {#1} } }
      { \prg_return_true: } { \prg_return_false: }
  }
```

(*End definition for* `\xeCJK_family_if_exist:xTF`.)

`\CJKfamily` 用于切换 CJK 字体族。

```
\NewDocumentCommand \CJKfamily { t+ t- m }
  {
    \xeCJK_if_blank_x:nTF {#3}
      {
        \IfBooleanF {#1} { \IfBooleanF {#2} { \use_none:nn } }
        \xeCJK_family_if_exist_use:x { \l_xeCJK_family_tl }
      }
      {
        \IfBooleanTF {#2} { \xeCJK_family_if_exist_use:x {#3} }
          {
            \xeCJK_family_if_exist:xTF {#3}
              {
                \tl_set:Nx \l_xeCJK_family_tl {#3}
                \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
                \IfBooleanT {#1} { \__xeCJK_family_use:x {#3} }
              }
              { \__xeCJK_family_unknown_warning:x {#3} }
          }
      }
    \tex_ignorespaces:D
  }
\cs_new_protected_nopar:Npn \xeCJK_switch_family:n #1
  {
    \xeCJK_family_if_exist:xTF {#1}
      {
        \tl_set:Nx \l_xeCJK_family_tl {#1}
        \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
      }
      { \__xeCJK_family_unknown_warning:x {#1} }
  }
```

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

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

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

(*End definition for* \l_xeCJK_family_tl.)

__xeCJK_gobble_CJKfamily:

```
\cs_new_protected_nopar:Npn \__xeCJK_gobble_CJKfamily:
  { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
\DeclareExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } {  }
```

(*End definition for* __xeCJK_gobble_CJKfamily:.)

xeCJK_family_if_exist_use:x

```
\cs_new_protected_nopar:Npn \xeCJK_family_if_exist_use:x #1
  {
    \xeCJK_family_if_exist:xTF {#1}
      { \__xeCJK_family_use:x {#1} }
      { \__xeCJK_family_unknown_warning:x {#1} }
  }
```

(*End definition for* \xeCJK_family_if_exist_use:x.)

__xeCJK_family_unknown_warning:n

```
\cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:n #1
  {
    \prop_if_empty:NF \g__xeCJK_family_font_name_prop
      {
        \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
          {
            \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
            \__xeCJK_warning:nx { CJKfamily-Unknown } {#1}
          }
      }
  }
\cs_generate_variant:Nn \__xeCJK_family_unknown_warning:n { x }
\seq_new:N \g__xeCJK_unknown_family_seq
\__xeCJK_msg_new:nn { CJKfamily-Unknown }
  {
    Unknown~CJK~family~`\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\\\
    Try~to~use~`\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
  }
\cs_new_nopar:Npn \__xeCJK_msg_def_family_map:n #1
  {
    \str_case_x:nnn {#1}
      {
        \CJKrmdefault { \token_to_str:N \setCJKmainfont }
        \CJKsfdefault { \token_to_str:N \setCJKsansfont }
        \CJKttdefault { \token_to_str:N \setCJKmonofont }
      }
      { \token_to_str:N \setCJKfamilyfont {#1} }
    [...]{...}
  }
\cs_new_nopar:Npn \__xeCJK_msg_family_map:n #1
  {
    \str_case_x:nnn {#1}
      {
        \CJKrmdefault { \token_to_str:N \CJKrmdefault }
        \CJKsfdefault { \token_to_str:N \CJKsfdefault }
        \CJKttdefault { \token_to_str:N \CJKttdefault }
      }
      {#1}
  }
```

(*End definition for* __xeCJK_family_unknown_warning:n.)

\setCJKmainfont
\setCJKsansfont
\setCJKmonofont

设置文档的 CJK 普通字体、无衬线和等宽字体。

```
\NewDocumentCommand \setCJKmainfont { O { } m }
  {
    \xeCJK_set_family:xxx { \CJKrmdefault } {#1} {#2}
```

```
    \normalfont
  }
\cs_new_eq:NN \setCJKromanfont \setCJKmainfont
\NewDocumentCommand \setCJKsansfont { O { } m }
  {
    \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2}
    \normalfont
  }
\NewDocumentCommand \setCJKmonofont { O { } m }
  {
    \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2}
    \normalfont
  }
```

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

```
\@onlypreamble \setCJKmainfont
\@onlypreamble \setCJKmathfont
\@onlypreamble \setCJKsansfont
\@onlypreamble \setCJKmonofont
\@onlypreamble \setCJKromanfont
```

`\setCJKfamilyfont` `\newCJKfontfamily` `\CJKfontspec`

分别用于预声明 CJK 字体和随机调用 CJK 字体。

```
\NewDocumentCommand \setCJKfamilyfont { m O { } m }
  { \xeCJK_set_family:xxx {#1} {#2} {#3} }
\NewDocumentCommand \newCJKfontfamily { o m O { } m }
  {
    \tl_set:Nx \l__xeCJK_tmp_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
    \cs_new_protected_nopar:Npx #2 { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
    \xeCJK_set_family:xxx { \l__xeCJK_tmp_tl } {#3} {#4}
  }
\NewDocumentCommand \CJKfontspec { O { } m }
  {
    \use:x { \xeCJK_fontspec:nn {#1} {#2} }
    \tex_ignorespaces:D
  }
```

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

`\xeCJK_fontspec:nn`

```
\cs_new_protected_nopar:Npn \xeCJK_fontspec:nn #1#2
  {
    \prop_get:NnNTF \g__xeCJK_fontspec_prop
      { CJKfontspec/#1/#2/id } \l_xeCJK_family_tl
      { \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl }
      {
        \__xeCJK_fontspec:xnn
          { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
          {#1} {#2}
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_fontspec:nnn #1#2#3
  {
    \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
    \xeCJK_set_family:nnn {#1} {#2} {#3}
    \xeCJK_switch_family:n {#1}
  }
\cs_generate_variant:Nn \xeCJK_fontspec:nn { VV }
\cs_generate_variant:Nn \__xeCJK_fontspec:nnn { x }
\prop_new:N \g__xeCJK_fontspec_prop
```

(*End definition for* `\xeCJK_fontspec:nn`.)

`\defaultCJKfontfeatures` `\addCJKfontfeatures`

分别用于设置 CJK 字体的默认属性和增加当前 CJK 字体的属性。

```
\clist_new:N \g__xeCJK_default_features_clist
\NewDocumentCommand \defaultCJKfontfeatures { m }
  { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
\@onlypreamble \defaultCJKfontfeatures
\NewDocumentCommand \addCJKfontfeatures { s O { } m }
  {
```

```
    \xeCJK_add_font_features:Nxx #1 {#2} {#3}
    \tex_ignorespaces:D
  }
\cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
```

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

`xeCJK_add_font_features:Nnn`

```
\cs_new_protected_nopar:Npn \xeCJK_add_font_features:Nnn #1#2#3
  {
    \prop_get:NVNTF \g__xeCJK_family_font_name_prop
      \l_xeCJK_family_tl \l__xeCJK_font_name_tl
      {
        \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
        \seq_map_inline:Nn \g__xeCJK_sub_key_seq
          { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
        \seq_clear:N \l__xeCJK_sub_key_seq
        \clist_clear:N \l__xeCJK_add_block_features_clist
        \clist_map_inline:nn {#2}
          {
            \seq_if_in:NnTF \g__xeCJK_sub_key_seq {##1}
              {
                \seq_put_right:Nn \l__xeCJK_sub_key_seq {##1}
                \__xeCJK_add_sub_class_features:n {##1}
              }
              { \__xeCJK_warning:nx { SubBlock-undefined } {##1} }
          }
        \bool_if:nT { #1 && \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
          {
            \seq_map_function:NN
              \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
          }
        \prop_get:NVNT \g__xeCJK_family_font_options_prop
          \l_xeCJK_family_tl \l__xeCJK_font_options_clist
          {
            \bool_if:nT
              { \seq_if_empty_p:N \l__xeCJK_sub_key_seq || #1 }
              {
                \clist_concat:NNN \l__xeCJK_font_options_clist
                  \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
              }
            \clist_concat:NNN \l__xeCJK_font_options_clist
              \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
          }
        \xeCJK_fontspec:VV \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
      }
      { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
  }
\clist_new:N \l__xeCJK_add_font_features_clist
\clist_new:N \l__xeCJK_add_block_features_clist
\cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nxx , Nnx }
\__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
  {
    \token_to_str:N \addCJKfontfeature (s)~ignored.\\\\
    It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
  }
```

(*End definition for* `\xeCJK_add_font_features:Nnn`*.*)

`\__xeCJK_add_sub_class_features:n`

```
\cs_new_protected_nopar:Npn \__xeCJK_add_sub_class_features:n #1
  {
    \prop_get:NoNTF \g__xeCJK_family_font_name_prop
      { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
      {
        \prop_get:NoN \g__xeCJK_family_font_options_prop
          { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
      }
      {
        \prop_get:NxNTF \g__xeCJK_family_font_name_prop
```

```
          { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
          {
            \prop_get:NxN \g__xeCJK_family_font_options_prop
              { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
          }
          {
            \prop_get:NVN \g__xeCJK_family_font_options_prop
              \l_xeCJK_family_tl \l__xeCJK_sub_font_options_clist
            \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl
          }
      }
    \clist_concat:NNN \l__xeCJK_sub_font_options_clist
      \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
    \clist_put_right:Nx \l__xeCJK_add_block_features_clist
      {
        #1 =
          {
            [ \exp_not:V \l__xeCJK_sub_font_options_clist ]
            { \exp_not:V \l__xeCJK_sub_font_name_tl }
          }
      }
  }
\cs_generate_variant:Nn \prop_get:NnN   { Nx }
\cs_generate_variant:Nn \prop_get:NnNTF { Nx }
```

(*End definition for* `\__xeCJK_add_sub_class_features:n`.)

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

```
\__xeCJK_at_end_preamble:n
  {
    \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
    \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
      {
        \tl_gset:Nx \CJKfamilydefault
          {
            \str_case_x:nnn { \familydefault }
              {
                { \rmdefault } { \exp_not:N \CJKrmdefault }
                { \sfdefault } { \exp_not:N \CJKsfdefault }
                { \ttdefault } { \exp_not:N \CJKttdefault }
              }
              { \CJKfamilydefault }
          }
      }
    \cs_undefine:N \__xeCJK_family_default_wrap:n
    \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
      { \__xeCJK_warning:nx { no-CJKfamily } { \CJKfamilydefault } }
      {
        \xeCJK_family_if_exist:xF { \CJKfamilydefault }
          {
            \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
            \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
              { \use:n }
              {
                \xeCJK_family_if_exist:xTF { \CJKrmdefault }
                  { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
              }
              {
                \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
                  {
                    \prop_map_break:n
                      { \tl_gset_rescan:Nnn \CJKfamilydefault { } {#1} }
                  }
              }
            \__xeCJK_warning:nxx { CJKfamilydefault-undefined }
              { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
          }
```

```
        \xeCJK_switch_family:n { \CJKfamilydefault }
        \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
      }
  }
\__xeCJK_msg_new:nn { no-CJKfamily }
  {
    It~seems~that~you~have~not~declare~a~CJKfamily.\\
    If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
    `\__xeCJK_msg_def_family_map:n {#1}'\\\\
    in~the~preamble~to~declare~the~default~CJKfamily.\\
  }
\__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
  {
    Undefined~CJK~default~family~`\__xeCJK_msg_family_map:n {#1}'~
    has~been~replaced~by~`\__xeCJK_msg_family_map:n {#2}'.\\\\
    Try~to~use~`\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
  }
```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```
\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。

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

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

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

```
\cs_new_protected_nopar:Npn \xeCJK_set_mathfont:
  {
    \xeCJK_family_if_exist:xTF { \c__xeCJK_math_tl }
      { \tl_set:Nx \l__xeCJK_tmp_tl { \c__xeCJK_math_tl } }
      {
        \xeCJK_family_if_exist:xTF { \CJKfamilydefault }
          { \tl_set:Nx \l__xeCJK_tmp_tl { \CJKfamilydefault } }
          { \use_none:nnnnn }
      }
    \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmp_tl \l__xeCJK_tmp_tl
      {
        \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmp_tl }
        \DeclareSymbolFont { \c__xeCJK_math_tl } { \c__xeCJK_encoding_tl }
          { \c__xeCJK_math_family_tl } { \mddefault } { \shapedefault }
        \cs_if_free:cF
          { \c__xeCJK_encoding_tl/\c__xeCJK_math_family_tl/\bfdefault/\shapedefault }
          {
            \SetSymbolFont { \c__xeCJK_math_tl } { bold } { \c__xeCJK_encoding_tl }
              { \c__xeCJK_math_family_tl } { \bfdefault } { \shapedefault }
          }
        \int_const:Nn \c_xeCJK_math_fam_int { \use:c { sym \c__xeCJK_math_tl } }
        \clist_concat:NNN \l__xeCJK_tmp_clist
          \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
        \clist_concat:NNN \l__xeCJK_tmp_clist
          \l__xeCJK_tmp_clist \c__xeCJK_FullRight_chars_clist
        \clist_map_inline:Nn \l__xeCJK_tmp_clist
          {
            \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
              { \c_zero } { \c_xeCJK_math_fam_int }
          }
      }
  }
```

(*End definition for* `\xeCJK_set_mathfont:`*.*)

\xeCJK_gset_mathcode:nnnn

```
\cs_new_protected_nopar:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
  {
    \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
    \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
      {
        \tex_global:D \xeCJK_xetex_mathcode:w
          \l__xeCJK_begin_int = #3 ~ #4 \l__xeCJK_begin_int
        \int_incr:N \l__xeCJK_begin_int
      }
  }
```

(*End definition for* `\xeCJK_gset_mathcode:nnnn`.)

5.15 抄录环境中的间距调整

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

```
\int_new:N \l__xeCJK_verb_case_int
\keys_define:nn { xeCJK / options }
  {
    Verb .choices:nn =
      { true , env+ , env , false }
      { \int_set_eq:NN \l__xeCJK_verb_case_int \l_keys_choice_int } ,
    Verb  .default:n = { env }
  }
\cs_new_protected_nopar:Npn \__xeCJK_verb_font_hook:
  {
    \if_case:w \l__xeCJK_verb_case_int
    \or:
      \__xeCJK_nobreak_skip_zero:
    \or:
      \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
        { \xeCJKVerbAddon }
        { \__xeCJK_nobreak_skip: }
    \or:
      \int_compare:nNnTF \etex_currentgrouptype:D = \c_fourteen
        { \xeCJKVerbAddon }
        { \__xeCJK_nobreak_skip_zero: }
    \fi:
  }
\__xeCJK_after_preamble:n
  {
    \cs_set_protected_nopar:Npx \verbatim@font
      { \exp_not:o { \verbatim@font } \__xeCJK_verb_font_hook: }
  }
```

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

__xeCJK_nobreak_skip_zero:
__xeCJK_nobreak_skip:

```
\cs_new_protected_nopar:Npn \__xeCJK_nobreak_skip_zero:
  {
    \__xeCJK_reset_shipout_skip:
    \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
    \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
    \tl_put_right:Nn \l__xeCJK_reset_shipout_skip_hook_tl
      {
        \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
        \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
      }
    \xeCJK_cs_clear:N \CJKglue
    \xeCJK_cs_clear:N \CJKecglue
    \xeCJK_cs_clear:N \xeCJK_check_for_glue:
    \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \xeCJK_class_group_end:
    \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
    \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
  }
\cs_new_protected_nopar:Npn \__xeCJK_nobreak_skip:
  {
```

```
    \__xeCJK_reset_shipout_skip:
    \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_ccglue_skip
    \skip_if_eq:nnTF { \l__xeCJK_ccglue_skip } { \c_zero_skip }
      { \xeCJK_cs_clear:N \CJKglue }
      { \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue: }
    \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
    \skip_if_eq:nnTF { \l__xeCJK_ecglue_skip } { \c_zero_skip }
      { \xeCJK_cs_clear:N \CJKecglue }
      { \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue: }
    \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
    \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
  }
\cs_new_protected_nopar:Npn \__xeCJK_nobreak_ccglue:
  { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ccglue_skip }
\cs_new_protected_nopar:Npn \__xeCJK_nobreak_ecglue:
  { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ecglue_skip }
\cs_new_protected_nopar:Npn \__xeCJK_nobreak_hskip:n
  { \xeCJK_no_break: \skip_horizontal:n }
```

(*End definition for* `\__xeCJK_nobreak_skip_zero:` *and* `\__xeCJK_nobreak_skip:`.)

`__xeCJK_reset_shipout_skip:`

```
\cs_new_protected_nopar:Npn \__xeCJK_reset_shipout_skip:
  {
    \cs_set_eq:NN \__xeCJK_shipout_CJKglue:   \CJKglue
    \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
    \cs_set_eq:NN \__xeCJK_shipout_punct_hskip:n \__xeCJK_punct_hskip:n
    \cs_set_eq:NN
      \__xeCJK_shipout_punct_breakable_kern:n \__xeCJK_punct_breakable_kern:n
    \tl_set:Nx \l__xeCJK_off_verb_addon_tl
      {
        \exp_not:c
          { bool_set_ \bool_if:NTF \l__xeCJK_xecglue_bool  { true } { false } :N }
          \l__xeCJK_xecglue_bool
        \exp_not:n
          {
            \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
            \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
            \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_shipout_punct_hskip:n
            \cs_set_eq:NN
              \__xeCJK_punct_breakable_kern:n \__xeCJK_shipout_punct_breakable_kern:n
            \l__xeCJK_reset_shipout_skip_hook_tl
          }
      }
    \__xeCJK_add_to_shipout:n { \l__xeCJK_off_verb_addon_tl }
    \xeCJKsetup { xCJKecglue = false }
  }
\tl_new:N \l__xeCJK_reset_shipout_skip_hook_tl
```

(*End definition for* `\__xeCJK_reset_shipout_skip:`.)

`\xeCJKOffVerbAddon`
`\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`。以字母“M”的宽度是否等于 `\fontdimen2` 来判断当前字体是否是等宽字体。如果不是等宽字体，则设置间距为零或正文间距。

```
\NewDocumentCommand \xeCJKVerbAddon { }
  {
    \int_compare:nNnF \etex_currentgrouplevel:D = \c_zero
      {
        \bool_if:NF \l__xeCJK_listings_env_bool
          {
            \dim_compare:nNnTF
              { \tex_fontdimen:D \c_two \tex_font:D } =
              { \etex_fontcharwd:D \tex_font:D \c__xeCJK_mono_letter_int }
              {
                \__xeCJK_set_verb_exspace:
                \__xeCJK_verb_addon:
              }
              {
```

```
                \int_if_odd:nTF { \l__xeCJK_verb_case_int }
                  { \__xeCJK_nobreak_skip_zero: }
                  { \__xeCJK_nobreak_skip: }
              }
          }
      }
  }
\int_const:Nn \c__xeCJK_mono_letter_int { 77 }
\bool_new:N \l__xeCJK_listings_env_bool
\NewDocumentCommand \xeCJKOffVerbAddon { }
  { \tl_use:N \l__xeCJK_off_verb_addon_tl }
\tl_new:N \l__xeCJK_off_verb_addon_tl
\cs_new_protected_nopar:Npn \__xeCJK_verb_addon:
  {
    \bool_if:NF \l__xeCJK_verb_addon_bool
      {
        \bool_set_true:N \l__xeCJK_verb_addon_bool
        \__xeCJK_set_char_class_eq:nn { FullLeft }    { CJK }
        \__xeCJK_set_char_class_eq:nn { FullRight }   { CJK }
        \__xeCJK_set_char_class_eq:nn { HalfLeft }    { Default }
        \__xeCJK_set_char_class_eq:nn { HalfRight }   { Default }
        \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
        \cs_set_eq:NN \__xeCJK_shipout_CJKglue:   \CJKglue
        \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
        \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
        \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
        \cs_set_protected_nopar:Npx \xeCJKOffVerbAddon
          {
            \__xeCJK_reset_char_class:n { FullLeft }
            \__xeCJK_reset_char_class:n { FullRight }
            \__xeCJK_reset_char_class:n { HalfLeft }
            \__xeCJK_reset_char_class:n { HalfLeft }
            \__xeCJK_reset_char_class:n { NormalSpace }
            \exp_not:c
              { bool_set_ \bool_if:NTF \l__xeCJK_xecglue_bool  { true } { false } :N }
              \l__xeCJK_xecglue_bool
            \exp_not:n
              {
                \cs_set_eq:NN \CJKglue   \__xeCJK_shipout_CJKglue:
                \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
                \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
                \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
              }
          }
        \__xeCJK_add_to_shipout:n { \xeCJKOffVerbAddon }
        \xeCJKsetup { xCJKecglue = false }
      }
    \skip_if_eq:nnTF { \l__xeCJK_verb_exspace_skip } { \c_zero_skip }
      {
        \xeCJK_cs_clear:N \CJKglue
        \xeCJK_cs_clear:N \CJKecglue
      }
      {
        \skip_set_eq:NN \l__xeCJK_ccglue_skip \l__xeCJK_verb_exspace_skip
        \skip_set:Nn \l__xeCJK_ecglue_skip { .5 \l__xeCJK_verb_exspace_skip }
        \cs_set_eq:NN \CJKglue   \__xeCJK_nobreak_ccglue:
        \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue:
      }
    \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
    \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_CJK_and_Boundary:w
  }
\cs_new_protected_nopar:Npn \__xeCJK_verb_CJK_and_Boundary:w
  { \xeCJK_class_group_end: \CJKecglue }
\cs_new_protected_nopar:Npn \__xeCJK_reset_char_class:n #1
  {
    \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
    \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
      { \XeTeXcharclass ##1 = \l__xeCJK_tmp_int }
  }
```

```
\bool_new:N \l__xeCJK_verb_addon_bool
\cs_new_eq:NN \CJKfixedspacing \xeCJKVerbAddon
```

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

`\__xeCJK_set_verb_exspace:` 在抄录环境中，CJK 文字之间的间距为当前西文字体两个空格的宽度与当前字体大小之差，而与西文和空格的间距为 CJK 文字之间的间距的一半。

```
\cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
  {
    \tl_if_exist:cTF { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
      {
        \skip_set:Nn \l__xeCJK_verb_exspace_skip
          { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
      }
      {
        \tl_set:Nx \l__xeCJK_current_coor_tl { \l_xeCJK_family_tl/\curr@fontshape }
        \prop_get:NVNTF \g__xeCJK_scale_family_prop
          \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
          {
            \tl_set_eq:NN \xeCJK@family \l_xeCJK_family_tl
            \skip_zero:N \l__xeCJK_verb_exspace_skip
          }
          {
            \group_begin: \xeCJK_select_font: \exp_after:wN \group_end:
            \exp_after:wN \__xeCJK_set_verb_exspace:n
            \exp_after:wN { \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00 }
          }
      }
  }
\skip_new:N \l__xeCJK_verb_exspace_skip
```

(*End definition for* `\__xeCJK_set_verb_exspace:`*.*)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:n #1
  {
    \skip_set:Nn \l__xeCJK_verb_exspace_skip
      { \c_two \tex_fontdimen:D \c_two \tex_font:D - #1 }
    \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
      {
        \skip_zero:N \l__xeCJK_verb_exspace_skip
        \use:x
          {
            \__xeCJK_set_verb_scale:nn
              { \dim_to_fp:n { \c_two \tex_fontdimen:D \c_two \tex_font:D } }
              { \dim_to_fp:n {#1} }
          }
      }
      {
        \tl_const:cx { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
          { \skip_use:N \l__xeCJK_verb_exspace_skip }
      }
  }
```

(*End definition for* `\__xeCJK_set_verb_exspace:n`*.*)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_set_verb_scale:nn #1#2
  {
    \fp_set:Nn \l__xeCJK_scale_factor_fp { #1 / #2 }
    \__xeCJK_warning:nxx { scale-factor }
      { \fp_eval:n { round - ( \l__xeCJK_scale_factor_fp , 4 ) } }
      { \fp_eval:n { round + ( #2 / #1 , 4 ) } }
    \xeCJK_add_font_features:Nnx \c_true_bool
      { } { Scale = { \fp_use:N \l__xeCJK_scale_factor_fp } }
    \prop_gput:NVV \g__xeCJK_scale_family_prop
      \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
  }
\__xeCJK_msg_new:nn { scale-factor }
  {
```

```
    `\token_to_str:N \xeCJKVerbAddon'~may~not~work~properly.\\\\
    You~may~set~`Scale=#1'~to~CJKfamily~
    `\__xeCJK_msg_family_map:n { \l_xeCJK_family_tl }',\\
    or~set~`Scale=#2'~to~family~
    `\str_if_eq_x:nnTF \f@family \ttdefault
      { \token_to_str:N \ttdefault } { \f@family }'.
  }
\fp_new:N \l__xeCJK_scale_factor_fp
\prop_new:N \g__xeCJK_scale_family_prop
```

(*End definition for* `\__xeCJK_set_verb_scale:nn`.)

`\xeCJK_visible_space:` 如果文档不使用 EU1 作为默认字体编码，那么默认的打字机字体族很可能是传统的 TEX 字体，这时可视空格按照 OT1 编码传统一般就是字体中的 `\char32`。这里加入 `\scan_stop:` 的目的是强制发生状态转移。这样当空格出现在 CJK 文字后面时，使字体回到西文，保证在当前西文字体而不是在 CJK 字体中检查有没有 U+2423。

```
\cs_new_protected_nopar:Npn \xeCJK_visible_space:
  {
    \bool_if:NT \l__xeCJK_CJK_group_bool { \scan_stop: }
    \xeCJK_glyph_if_exist:NTF { ^^^^2423 }
      { ^^^^2423 }
      {
        \int_compare:nNnTF { \XeTeXfonttype \tex_font:D } = \c_zero
          {
            \str_if_eq_x:nnTF { \f@family } { \ttdefault }
              { \c_catcode_other_space_tl }
              { \textvisiblespace }
          }
          { \xeCJK_visible_space_fallback: }
      }
  }
\AtEndOfPackage
  { \cs_gset_eq:NN \fontspec_visible_space: \xeCJK_visible_space: }
```

(*End definition for* `\xeCJK_visible_space:`.)

`\xeCJK_visible_space_fallback:` fontspec 使用 lmtt 字体中的可视空格符号(U+2423)作为当前字体中相应符号的后备。但是 lmtt 的字体大小未必与当前字体匹配。因此，我们在这里做一些调整，以保证使用后备可视空格符号时，也能保证对齐。

```
\cs_new_protected_nopar:Npn \xeCJK_visible_space_fallback:
  { {
      \cs_if_exist_use:cF { xeCJK/space/\curr@fontshape/\f@size }
        { \xeCJK_set_visible_space_font: }
      ^^^^2423
  } }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_set_visible_space_font:
  {
    \tl_set:Nx \l__xeCJK_current_coor_tl { xeCJK/space/\curr@fontshape/\f@size }
    \exp_after:wN \__xeCJK_set_visible_space_size:n
    \exp_after:wN { \dim_use:N \tex_fontdimen:D \c_two \tex_font:D }
    \xeCJK_font_gset_to_current:c { \l__xeCJK_current_coor_tl }
  }
\cs_new_protected_nopar:Npn \__xeCJK_set_visible_space_size:n #1
  {
    \fontencoding { \g_fontspec_encoding_tl }
    \tl_set:Nx \f@family { lmtt }
    \selectfont
    \dim_compare:nNnF {#1} = { \tex_fontdimen:D \c_two \tex_font:D }
      {
        \fontsize
          {
            \dim_eval:n
              {
                \f@size pt *
```

```
                \dim_ratio:nn {#1} { \tex_fontdimen:D \c_two \tex_font:D }
              }
          }
          { \f@baselineskip }
        \selectfont
      }
  }
```

(*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。

```
\keys_define:nn { xeCJK / options }
  {
    LocalConfig .choice: ,
    LocalConfig / false   .code:n =
      { \bool_gset_false:N \g__xeCJK_config_bool } ,
    LocalConfig / true    .code:n =
      {
        \bool_gset_true:N \g__xeCJK_config_bool
        \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
      } ,
    LocalConfig / unknown .code:n =
      {
        \bool_gset_true:N \g__xeCJK_config_bool
        \tl_gset:Nx \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
      } ,
    LocalConfig        .default:n = { true }
  }
\tl_new:N \g__xeCJK_config_name_tl
\bool_new:N \g__xeCJK_config_bool
```

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

indentfirst 首行是否缩进。

```
\keys_define:nn { xeCJK / options }
  {
    CJKnumber    .bool_gset:N = \g__xeCJK_number_bool ,
    indentfirst  .bool_gset:N = \g__xeCJK_indent_bool ,
    normalindentfirst .meta:n = { indentfirst = false }
  }
```

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

quiet silent 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。

```
\keys_define:nn { xeCJK / options }
  {
    quiet .code:n =
      {
        \msg_redirect_module:nnn { xeCJK } { warning } { info }
        \msg_redirect_module:nnn { xeCJK } { info }    { none }
        \xeCJK_if_package_loaded:nF { fontspec }
          { \PassOptionsToPackage { quiet } { fontspec } }
      } ,
    silent .code:n =
      {
        \msg_redirect_module:nnn { xeCJK } { warning } { none }
        \msg_redirect_module:nnn { xeCJK } { info }    { none }
        \xeCJK_if_package_loaded:nF { fontspec }
          { \PassOptionsToPackage { silent } { fontspec } }
      } ,
    unknown .code:n =
      {
        \xeCJK_if_package_loaded:nTF { fontspec }
          { \__xeCJK_error:nx { key-unknown } { \l_keys_key_tl } }
          { \PassOptionsToPackage { \l_keys_key_tl } { fontspec } }
      }
  }
```

```
\__xeCJK_msg_new:nn { key-unknown }
  {
    Sorry,~but~\l__keys_module_tl \ does~not~have~a~key~called~`#1'.\\\\
    The~key~`#1'~is~being~ignored.
  }
```

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

5.17 xeCJK 初始化设置

\CJKsymbol
\CJKpunctsymbol

```
\cs_new_nopar:Npn \CJKsymbol      #1 {#1}
\cs_new_nopar:Npn \CJKpunctsymbol #1 {#1}
```

(*End definition for* `\CJKsymbol` *and* `\CJKpunctsymbol`.)

xeCJK 宏包的初始化设置。

```
\keys_set:nn { xeCJK / options }
  {
   CJKglue      = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
    CJKecglue       = { ~ } ,
    xCJKecglue      = false ,
    CheckSingle     = false ,
    PlainEquation   = false ,
    CheckFullRight  = false ,
    CJKspace        = false ,
    CJKmath         = false ,
    CJKnumber       = false ,
    xeCJKactive     = true  ,
    LocalConfig     = true  ,
    indentfirst     = true  ,
    Verb            = env   ,
    EmboldenFactor  = 4     ,
    SlantFactor     = 0.167 ,
    PunctStyle      = quanjiao ,
    NewLineCS       = { \par \[ } ,
    EnvCS           = { \begin \end } ,
    NoBreakCS       = { \footnote \footnotemark \nobreak } ,
    KaiMingPunct    = { ^^^^3002 ^^^^ff0e ^^^^ff1f ^^^^ff01 } ,
    LongPunct       = { ^^^^2014 ^^^^2015 ^^^^2500 ^^^^2025 ^^^^2026 } ,
   MiddlePunct    = { ^^^^2014 ^^^^2015 ^^^^2027 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
    AllowBreakBetweenPuncts = false
  }
\defaultCJKfontfeatures { Script = CJK }
```

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

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

\c__xeCJK_encoding_tl

保存 fontspec 声明字体时使用的字体编码。

```
\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。

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

对不能通过 `\xeCJKsetup` 设置的选项给出警告。

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

\CJKrmdefault
\CJKsfdefault
\CJKttdefault
\CJKfamilydefault

```
\tl_if_exist:NF \CJKrmdefault { \tl_gset:Nn \CJKrmdefault { rm } }
\tl_if_exist:NF \CJKsfdefault { \tl_gset:Nn \CJKsfdefault { sf } }
\tl_if_exist:NF \CJKttdefault { \tl_gset:Nn \CJKttdefault { tt } }
\tl_new:N \l__xeCJK_family_default_init_tl
\cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
\tl_set:Nx \l__xeCJK_family_default_init_tl
  {
    \exp_not:N \__xeCJK_family_default_wrap:n
      {
        \tl_if_exist:NTF \CJKfamilydefault
          { \exp_not:V \CJKfamilydefault }
          { \exp_not:N \CJKrmdefault }
      }
  }
\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。

```
\NewDocumentCommand \xeCJKsetup { +m }
  {
    \keys_set:nn { xeCJK / options } {#1}
    \tex_ignorespaces:D
  }
```

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

\xeCJKsetemboldenfactor
\xeCJKsetslantfactor

```
\NewDocumentCommand \xeCJKsetemboldenfactor { m }
  { \xeCJKsetup { EmboldenFactor = {#1} } }
\NewDocumentCommand \xeCJKsetslantfactor { m }
  { \xeCJKsetup { SlantFactor = {#1} } }
```

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

\punctstyle
\xeCJKplainchr

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

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

\CJKsetecglue

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

(*End definition for* `\CJKsetecglue`.)

\CJKspace
\CJKnospace

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

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

\xeCJKallowbreakbetweenpuncts
\xeCJKnobreakbetweenpuncts

```
\NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
  { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
\NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
  { \xeCJKsetup { AllowBreakBetweenPuncts = false } }
```

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

\xeCJKenablefallback
\xeCJKdisablefallback

```
\NewDocumentCommand \xeCJKenablefallback { }
  { \xeCJKsetup { AutoFallBack = true } }
\NewDocumentCommand \xeCJKdisablefallback { }
  { \xeCJKsetup { AutoFallBack = false } }
```

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

\xeCJKsetcharclass

```
\NewDocumentCommand \xeCJKsetcharclass { m m m }
  {
    \xeCJK_set_char_class:nnn {#1} {#2} {#3}
    \xeCJKResetPunctClass
  }
```

(*End definition for* \xeCJKsetcharclass.)

5.18 兼容性修补

__xeCJK_update_url_font:
\Url@MathSetup

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

```
\cs_new_protected_nopar:Npn \__xeCJK_update_url_font:
  {
    \group_begin: \xeCJK_select_font: \exp_after:wN \group_end:
    \exp_after:wN \tex_textfont:D \exp_after:wN \c_xeCJK_math_fam_int
    \tex_the:D \tex_font:D
  }
\__xeCJK_after_end_preamble:n
  {
    \bool_if:nT { \g__xeCJK_math_bool && \cs_if_exist_p:N \Url@MathSetup }
      { \tl_put_right:Nn \Url@MathSetup { \__xeCJK_update_url_font: } }
  }
```

(*End definition for* __xeCJK_update_url_font: *and* \Url@MathSetup.)

\fontspec_setup_maths:
\mathrm

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

```
\cs_if_free:NF \fontspec_setup_maths:
  {
    \cs_gset_protected_nopar:Npx \fontspec_setup_maths:
      {
        \exp_not:o
          {
            \fontspec_setup_maths:
            \tl_if_empty:NT \g_fontspec_bfmathrm_tl
              { \DeclareSymbolFontAlphabet \mathrm { operators } }
          }
      }
  }
```

(*End definition for* \fontspec_setup_maths: *and* \mathrm.)

\(
\)
\math
\endmath
\ensuremath
__xeCJK_math_robust:N

\(的在 LaTeX 2ε 中的定义是

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

这个定义最开始的 \relax 是为了防止 \(出现在表格单元格的开始位置时，模式判断不正确（因为 TeX 会先看单元格中第一个不可展的非空格记号是否是 \omit 或 \noalign）。但是它会造成一个边界，使 xeCJK 不能看到 \relax 后面出现的 $，从而不能加入间距[10]。使用 ε-TeX 的 \protected 来定义它，可以不需要 \relax，或者将 \relax 改成 \scan_align_safe_stop:，都可以避免这些情况。同时 fixltx2e 中还使用了 \MakeRobust\(，我们需要小心处理。另外 ulem 也定义了一个 \MakeRobust，如果它被放在 fixltx2e 之前载入，那么 fixltx2e 的定义就会失效（因为 fixltx2e 使用 \providecommand* 来定义 \MakeRobust）。但是 ulem 的定义并不完全正确，没有考虑 TeX 不会略去控制符号后面的空格的情况。

```
\cs_new_protected_nopar:Npn \__xeCJK_math_robust:N #1
  { \exp_args:NNc \__xeCJK_math_robust_aux:NN #1 { \cs_to_str:N #1 ~ } }
\cs_new_protected_nopar:Npn \__xeCJK_math_robust_aux:NN #1#2
  {
    \exp_args:Nx \str_case:nnTF { \token_get_replacement_spec:N #1 }
      {
        { \x@protect #1 \protect #2 } { }
        { \protect #2 } { }
      }
      { \__xeCJK_math_robust:NN #1#2 }
```

[10] http://tex.stackexchange.com/q/124773

```
      { \__xeCJK_math_robust:NN #1#1 }
  }
\cs_new_protected_nopar:Npn \__xeCJK_math_robust:NN #1#2
  {
    \str_if_eq_x:nnTF { \token_get_arg_spec:N #2 } { }
      {
        \exp_args:No \tl_if_head_eq_meaning:nNTF {#2} \scan_stop:
          {
            \cs_gset_protected_nopar:Npx #1
              { \scan_align_safe_stop: \tl_tail:N #2 }
          }
          {
            \cs_if_eq:NNTF #1 \ensuremath
              {
                \cs_gset_protected_nopar:Npx #1
                  { \scan_align_safe_stop: \exp_not:o {#2} }
              }
              {
                \__xeCJK_warning:nxx { robust-failure }
                  { \token_to_str:N #1 } { \token_to_meaning:N #2 }
              }
          }
      }
      {
        \__xeCJK_warning:nxx { robust-failure }
          { \token_to_str:N #1 } { \token_to_meaning:N #2 }
      }
  }
\__xeCJK_msg_new:nnn { robust-failure }
  { xeCJK~can~not~make~`#1'~robust. }
  {
    The~current~meaning~of~`#1'~is:\\
    \iow_indent:n {#2}
  }
\__xeCJK_math_robust:N \(
\__xeCJK_math_robust:N \)
\__xeCJK_math_robust:N \math
\__xeCJK_math_robust:N \endmath
\__xeCJK_math_robust:N \ensuremath
```

(*End definition for* `\(` *and others.*)

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

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

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

```
\bool_if:nF
  {
    \xeCJK_if_package_loaded_p:n { amsmath } ||
    \xeCJK_if_package_loaded_p:n { amsthm }
  }
  {
    \__xeCJK_math_robust:N \[
    \__xeCJK_math_robust:N \]
  }
```

(*End definition for* `\[` *and* `\]`.)

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

```
\UndeclareTextCommand \nobreakspace { \UTFencname }
\RenewDocumentCommand \nobreakspace { } { \leavevmode \nobreak \ }
```

(*End definition for* *nobreakspace*.)

当符号命令紧跟在 CJK 字符类后面时，强制发生状态转移，使字体回到西文状态。

```
\AtBeginUTFCommand { \bool_if:NT \l__xeCJK_CJK_group_bool { \scan_stop: } }
```

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

```
\__xeCJK_msg_new:nn { conflict-package }
  {
    The~`#1'~package~is~too~old. \\
    Please~update~an~up~to~date~version~of~it\\
    using~your~TeX~package~manager~or~from~CTAN.
  }
\xeCJK_if_package_loaded:nTF { realscripts }
  {
    \@ifpackagelater { realscripts } { 2010/10/10 } { }
      {
        \__xeCJK_error:nx { conflict-package }
          {
            \xeCJK_if_package_loaded:nTF { xltxtra }
              { xltxtra } { realscripts }
          }
      }
  }
  {
    \cs_new_eq:NN \__xeCJK_dim_max:nn \dim_max:nn
    \cs_new_eq:NN \__xeCJK_dim_min:nn \dim_min:nn
    \__xeCJK_at_end_preamble:n
      {
        \xeCJK_if_package_loaded:nT { realscripts }
          {
            \@ifpackagelater { realscripts } { 2010/10/10 } { }
              {
                \cs_gset_eq:NN \dim_max:nn \__xeCJK_dim_max:nn
                \cs_gset_eq:NN \dim_min:nn \__xeCJK_dim_min:nn
              }
          }
        \cs_undefine:N \__xeCJK_dim_max:nn
        \cs_undefine:N \__xeCJK_dim_min:nn
      }
  }
```

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

```
\RenewDocumentCommand \fontfamily { m }
  {
    \tl_set:Nx \f@family {#1}
    \__xeCJK_update_family:nn {#1}
      {
        { \rmdefault }     { \xeCJK_switch_family:n { \CJKrmdefault } }
        { \sfdefault }     { \xeCJK_switch_family:n { \CJKsfdefault } }
        { \ttdefault }     { \xeCJK_switch_family:n { \CJKttdefault } }
        { \familydefault } { \xeCJK_switch_family:n { \CJKfamilydefault } }
      }
  }
\cs_new_eq:NN \__xeCJK_update_family:nn \str_case:nn
```

(*End definition for* *fontfamily*.)

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

```
\cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
\tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
\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。

```
\cs_new_protected_nopar:Npn \xeCJK@italiccorr
  {
    \int_compare:nNnTF \XeTeXinterchartokenstate > \c_zero
      {
        \xeCJK_if_last_node:nTF { default }
          {
            \xeCJK_remove_node: \@@italiccorr
            { \xeCJK_make_node:n { default } }
          }
          {
            \xeCJK_if_last_node:nTF { CJK }
              {
                \xeCJK_remove_node: \@@italiccorr
                { \xeCJK_make_node:n { CJK } } \use:n
              }
              {
                \xeCJK_if_last_node:nTF { CJK-space }
                  {
                    \xeCJK_remove_node: \@@italiccorr
                    { \xeCJK_make_node:n { CJK-space } } \use:n
                  }
                  { \@@italiccorr \use_none:n }
              }
```

\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:)。

```
              {
                                \exp_after:wN \exp_after:wN \exp_after:wN
                \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
                \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
                \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
                \xeCJK_ignore_spaces:w
              }
          }
      }
      { \@@italiccorr }
  }
```

(*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。

```
\__xeCJK_after_end_preamble:n
  {
    \int_compare:nNnF
      { \c_three + \seq_count:N \g__xeCJK_new_class_seq } = \xe@alloc@intercharclass
      {
        \int_step_inline:nnnn \c_four \c_one \xe@alloc@intercharclass
          {
            \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
              { \__xeCJK_set_others_toks:n {#1} }
          }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_set_others_toks:n #1
  {
    \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
    \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
      {
        \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
        \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
        \xeCJK_app_inter_class_toks:nnx {##1} { Others }
          { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
        \xeCJK_pre_inter_class_toks:nnx { Others } {##1}
          { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
        \xeCJK_if_blank_x:nT
```

```
          { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
          {
            \xeCJK_copy_inter_class_toks:nnnn
              { Others } { Boundary } { Default } { Boundary }
          }
        \xeCJK_if_blank_x:nT
          { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
          {
            \xeCJK_copy_inter_class_toks:nnnn
              { Boundary } { Others } { Boundary } { Default }
          }
      }
  }
```

(*End definition for* `\__xeCJK_set_others_toks:n`.)

`\__xeCJK_group_begin:` `\__xeCJK_group_end:` 用于保护下面歧义宽度标点的分组。

```
\cs_new_eq:NN \__xeCJK_group_begin: \group_begin:
\cs_new_eq:NN \__xeCJK_group_end:   \group_end:
```

(*End definition for* `\__xeCJK_group_begin:` *and* `\__xeCJK_group_end:`.)

`\textellipsis` 单独处理宽度有分歧的几个标点：包括省略号、破折号、间隔号、引号等中西文混用的符号，保证其命令形式输出的是西文字体。

```
\tl_map_inline:nn
  {
    \textellipsis  \textemdash     \textperiodcentered \textcentereddot
    \textquoteleft \textquoteright \textquotedblleft   \textquotedblright
  }
  {
    \AtBeginUTFCommand [#1] { \__xeCJK_group_begin: \makexeCJKinactive }
    \AtEndUTFCommand   [#1] { \__xeCJK_group_end: }
  }
```

(*End definition for* `\textellipsis`.)

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

```
\__xeCJK_after_preamble:n
  {
    \tl_put_left:Nn \tipaencoding { \makexeCJKinactive }
    \cs_new_eq:NN \__xeCJK_aux_r:n \r
    \cs_set_nopar:Npn \r #1
      {
        \bool_if:nTF
          {
            \str_if_eq_x_p:nn { \f@encoding } { T1 } &&
            \str_if_eq_x_p:nn {#1} { u }
          }
          { { \makexeCJKinactive \__xeCJK_aux_r:n {#1} } }
          { \__xeCJK_aux_r:n {#1} }
      }
    \xeCJK_if_package_loaded:nT { pifont }
      {
        \RenewDocumentCommand \Pifont { m }
          { \makexeCJKinactive \usefont { U } {#1} { m } { n } }
      }
  }
```

简单处理与 hyperref 宏包的兼容问题。

```
\__xeCJK_after_end_preamble:n
  {
    \xeCJK_if_package_loaded:nT { hyperref }
      {
        \pdfstringdefDisableCommands
          {
            \__xeCJK_gobble_CJKfamily:
            \xeCJK_cs_clear:N \makexeCJKinactive
            \xeCJK_cs_clear:N \__xeCJK_group_begin:
            \xeCJK_cs_clear:N \__xeCJK_group_end:
          }
      }
  }
```

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

```
\__xeCJK_after_end_preamble:n
  {
    \bool_if:nT
      { \xeCJK_if_package_loaded_p:n { cprotect } && \cs_if_exist_p:N \icprotect }
      { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
  }
```

\xeCJKcaption 可以使用 CJK 宏包中的 .cpx 文件。

```
\cs_if_exist:NF \CJK@ifundefined
  { \cs_set_eq:NN \CJK@ifundefined \cs_if_free:NTF }
\NewDocumentCommand \xeCJKcaption { o m }
  {
    \IfNoValueF {#1} { \XeTeXdefaultencoding "#1" }
    \use:x
      {
        \char_set_catcode_letter:n { 64 }
        \file_input:n { #2.cpx }
        \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
      }
    \XeTeXdefaultencoding "UTF-8"
  }
```

(*End definition for* \xeCJKcaption.)

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

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

```
\cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
  { \AtEndOfPackage { \RequirePackage { xeCJKfntef } } }
  {
    \__xeCJK_at_end_preamble:n
      {
        \xeCJK_if_package_loaded:nF { xeCJKfntef }
          {
            \xeCJK_if_package_loaded:nTF { CJKfntef }
              { \RequirePackage { xeCJKfntef } }
              {
                \xeCJK_if_package_loaded:nT { ulem }
                  { \RequirePackage { xeCJKfntef } }
              }
          }
      }
  }
```

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

```
\__xeCJK_at_end_preamble:n
  {
    \xeCJK_if_package_loaded:nT { listings }
      { \RequirePackage { xeCJK-listings } }
  }
```

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

```
\cs_new_protected:Npn \CJKaddEncHook #1#2
  {
    \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
      {
        \group_begin:
        \cs_set_nopar:Npn \Unicode ##1##2
          { (##1) * \c_two_hundred_fifty_six + (##2) }
        \cs_set_eq:NN \def \xeCJK_char_from_charcode:Nn
        #2
        \group_end:
        \tl_gset:Nn \CJK@tenthousand    { ^^^^4e07 }
        \tl_gset:Nn \CJK@hundredmillion { ^^^^4ebf }
      }
  }
\cs_new_protected_nopar:Npn \xeCJK_char_from_charcode:Nn #1#2
```

```
  {
    \group_begin:
    \char_set_lccode:nn { "4E00 } {#2}
    \tl_to_lowercase:n
      {
        \group_end:
        \tl_const:Nn #1 { ^^^^4e00 }
      }
  }
\bool_if:NT \g__xeCJK_number_bool { \RequirePackage { CJKnumb } }
```

(*End definition for* \CJKaddEncHook.)

最后引入本地配置文件。

```
\bool_if:NT \g__xeCJK_config_bool
  {
    \tl_const:Nn \c__xeCJK_config_ext_tl { cfg }
    \@onefilewithoptions
      { \g__xeCJK_config_name_tl } [ ] [ ] { \c__xeCJK_config_ext_tl }
  }
```

⟨/package⟩

5.19 xeCJKfntef

⟨*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。

```
\PassOptionsToPackage { normalem } { ulem }
\DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
\ProcessOptions \scan_stop:
```

```
\RequirePackage { xeCJK }
\RequirePackage { ulem }
\RequirePackage { CJKfntef }
\RequirePackage { environ }
```

```
\addto@hook \UL@hook { \xeCJK_hook_for_ulem: }
```

\xeCJK_hook_for_ulem:

```
\cs_new_protected_nopar:Npn \xeCJK_hook_for_ulem:
  {
    \bool_if:NF \l__xeCJK_ulem_hook_used_bool
      {
        \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
        \xeCJKsetup { CheckFullRight = false , xCJKecglue = false }
        \bool_if:NTF \l__xeCJK_ulem_skip_punct_bool
          { \cs_set_eq:NN \__xeCJK_ulem_leader_type: \UL@leadtype }
          {
            \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
            \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
          }
        \__xeCJK_ulem_initial:
        \xeCJK_glue_to_skip:nN
          {
            \cs_set_eq:NN \  \tex_space:D
            \cs_set_eq:NN \penalty \tex_penalty:D
            \cs_set_eq:NN \hskip \skip_horizontal:N
            \CJKglue
          } \l__xeCJK_ccglue_skip
        \xeCJK_glue_to_skip:nN
          {
            \cs_set_eq:NN \  \tex_space:D
            \cs_set_eq:NN \penalty \tex_penalty:D
            \cs_set_eq:NN \hskip \skip_horizontal:N
            \CJKecglue
          } \l__xeCJK_ecglue_skip
        \cs_set_protected_nopar:Npn \CJKglue
          { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
        \cs_set_protected_nopar:Npn \CJKecglue
          { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
      }
```

```
  }
\bool_new:N \l__xeCJK_ulem_hook_used_bool
```

(*End definition for* \xeCJK_hook_for_ulem:.)

\CJK@UL
\CJK@@UL

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

```
\cs_set_eq:NN \CJK@UL \CJK@@UL
\tl_replace_once:Nnn \CJK@UL { \ULon }
  { \bool_set_true:N \l__xeCJK_ulem_skip_punct_bool \ULon }
\tl_replace_once:Nnn \CJK@@UL { \ULon }
  { \bool_set_false:N \l__xeCJK_ulem_skip_punct_bool \ULon }
\bool_new:N \l__xeCJK_ulem_skip_punct_bool
```

(*End definition for* \CJK@UL *and* \CJK@@UL.)

__xeCJK_ulem_skip_punct_begin:
__xeCJK_ulem_skip_punct_end:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_begin:
  { \xeCJK_cs_clear:N \UL@leadtype }
\cs_new_protected_nopar:Npn \__xeCJK_ulem_skip_punct_end:
  { \cs_set_eq:NN \UL@leadtype \__xeCJK_ulem_leader_type: }
\xeCJK_cs_clear:N \__xeCJK_ulem_leader_type:
```

(*End definition for* __xeCJK_ulem_skip_punct_begin: *and* __xeCJK_ulem_skip_punct_end:.)

__xeCJK_ulem_initial:

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

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_initial:
  {
    \__xeCJK_ulem_swap_cs:NN
    \xeCJK_FullLeft_and_Default:  \__xeCJK_ulem_FullLeft_and_Default:
    \xeCJK_FullLeft_and_CJK:      \__xeCJK_ulem_FullLeft_and_CJK:
    \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
    \xeCJK_FullRight_and_CJK:     \__xeCJK_ulem_FullRight_and_CJK:
    \xeCJK_CJK_and_CJK:N          \__xeCJK_ulem_CJK_and_CJK:N
    \xeCJK_Boundary_and_Default:  \__xeCJK_ulem_Boundary_and_Default:
    \xeCJK_Boundary_and_NormalSp: \__xeCJK_ulem_Boundary_and_NormalSp:
    \xeCJK@fix@penalty            \__xeCJK_ulem_fix_penalty:
    \__xeCJK_punct_kern:n                 \__xeCJK_ulem_punct_kern:n
    \__xeCJK_punct_hskip:n                \__xeCJK_ulem_punct_hskip:n
    \__xeCJK_punct_breakable_kern:n       \__xeCJK_ulem_punct_breakable_kern:n
    \__xeCJK_CJK_and_Boundary_aux:        \__xeCJK_ulem_CJK_and_Boundary_aux:
    \__xeCJK_Default_and_FullLeft_glue:N  \__xeCJK_ulem_Default_and_FullLeft_glue:N
    \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N
    \__xeCJK_CJK_and_FullLeft_glue:N      \__xeCJK_ulem_CJK_and_FullLeft_glue:N
    \__xeCJK_CJK_and_FullRight_glue:N     \__xeCJK_ulem_CJK_and_FullRight_glue:N
    \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
    \q_recursion_tail \q_nil \q_recursion_stop
    \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
      {
        \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
          {
            \str_if_eq:nnTF {##1} {####1}
              {
                \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
                  { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
                \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
                  { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
              }
              {
                \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
                  { \__xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
              }
          }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_ulem_swap_cs:NN #1#2
  {
    \quark_if_recursion_tail_stop:N #1
    \xeCJK_swap_cs:NN #1#2
    \__xeCJK_ulem_swap_cs:NN
  }
```

(*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。

```
\cs_new_nopar:Npn \xeCJK_if_ulem_patch:TF
  {
    \if_meaning:w \  \LA@space
      \exp_after:wN \use_ii:nn
    \else:
      \exp_after:wN \use_i:nn
    \fi:
  }
```

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

__xeCJK_ulem_Boundary_and_Default:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_Default:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_if_last_node:nTF { CJK }
          { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
         { \xeCJK_if_last_node:nT { CJK-space } { \xeCJK_remove_node: \c_space_tl } }
      }
      { \__xeCJK_ulem_Boundary_and_Default: }
  }
```

(*End definition for* __xeCJK_ulem_Boundary_and_Default:.)

__xeCJK_ulem_Boundary_and_NormalSp:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_NormalSp:
  {
    \xeCJK_if_ulem_patch:TF
      { \xeCJK_if_last_node:nT { CJK-space } { \xeCJK_remove_node: \c_space_tl } }
      { \__xeCJK_ulem_Boundary_and_NormalSp: }
  }
```

(*End definition for* __xeCJK_ulem_Boundary_and_NormalSp:.)

__xeCJK_ulem_CJK_and_Boundary_aux:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_Boundary_aux:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_class_group_end:
        \UL@stop \__xeCJK_ulem_hskip:n { \c_zero_skip } \UL@start
        { \xeCJK_make_node:n { CJK } }
      }
      { \__xeCJK_ulem_CJK_and_Boundary_aux: }
  }
```

(*End definition for* __xeCJK_ulem_CJK_and_Boundary_aux:.)

__xeCJK_ulem_fix_penalty:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_fix_penalty:
  {
    \xeCJK_if_ulem_patch:TF
      { \fix@penalty }
      { \__xeCJK_ulem_fix_penalty: }
  }
```

(*End definition for* __xeCJK_ulem_fix_penalty:.)

__xeCJK_ulem_CJK_and_CJK:N

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_CJK:N
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_class_group_end:
        \UL@stop \__xeCJK_ulem_ccglue: \UL@start
        \__xeCJK_ulem_class_group_begin:
```

```
        \CJKsymbol
      }
      { \__xeCJK_ulem_CJK_and_CJK:N }
  }
```

(*End definition for* `\__xeCJK_ulem_CJK_and_CJK:N`.)

`\__xeCJK_ulem_class_group_begin:`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_class_group_begin:
  {
    \xeCJK_class_group_begin:
    \xeCJK_clear_Boundary_and_CJK_toks:
    \xeCJK_select_font:
  }
```

(*End definition for* `\__xeCJK_ulem_class_group_begin:`.)

`\__xeCJK_ulem_between_CJK_blocks:nnN`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_between_CJK_blocks:nnN #1#2
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_class_group_end:
        \UL@stop \__xeCJK_ulem_ccglue: \UL@start
        \xeCJK_class_group_begin:
        \xeCJK_clear_Boundary_and_CJK_toks:
        \__xeCJK_switch_font:nn {#1} {#2}
        \CJKsymbol
      }
      {
        \skip_horizontal:N \l__xeCJK_ccglue_skip
        \__xeCJK_switch_font:nn {#1} {#2}
        \CJKsymbol
      }
  }
```

(*End definition for* `\__xeCJK_ulem_between_CJK_blocks:nnN`.)

`\__xeCJK_ulem_Default_and_FullLeft_glue:N`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
  {
    \xeCJK_if_ulem_patch:TF
      {
        \UL@stop
        \__xeCJK_ulem_skip_punct_begin:
        \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
        \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
        \UL@start
      }
      { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
  }
```

(*End definition for* `\__xeCJK_ulem_Default_and_FullLeft_glue:N`.)

`__xeCJK_ulem_Boundary_and_FullLeft_glue:N`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
  {
    \xeCJK_if_ulem_patch:TF
      {
        \UL@stop
        \__xeCJK_ulem_skip_punct_begin:
        \int_compare:nNnTF \etex_lastnodetype:D = \c_one
          { \__xeCJK_zero_glue: }
          { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
        \UL@start
      }
      { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
  }
```

(*End definition for* `\__xeCJK_ulem_Boundary_and_FullLeft_glue:N`.)

__xeCJK_ulem_CJK_and_FullLeft_glue:N

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_class_group_end:
        \UL@stop
        \__xeCJK_ulem_skip_punct_begin:
        \__xeCJK_ulem_ccglue:
        \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
        \__xeCJK_punct_offset:NN \c__xeCJK_left_tl {#1}
        \UL@start
        \__xeCJK_ulem_class_group_begin:
      }
      { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
  }
```

(*End definition for* `\__xeCJK_ulem_CJK_and_FullLeft_glue:N`.)

__xeCJK_ulem_Default_and_FullRight_glue:N

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
  {
    \xeCJK_if_ulem_patch:TF
      {
        \UL@stop
        \__xeCJK_ulem_skip_punct_begin:
        \__xeCJK_punct_if_long:NTF {#1}
          { \__xeCJK_ulem_ccglue: }
          {
            \__xeCJK_punct_if_middle:NTF {#1}
              {
                \xeCJK_no_break:
                \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
                \__xeCJK_punct_bound_rule:NN \c__xeCJK_left_tl {#1}
              }
              { \xeCJK_no_break: }
          }
        \UL@start
      }
      { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
  }
```

(*End definition for* `\__xeCJK_ulem_Default_and_FullRight_glue:N`.)

__xeCJK_ulem_CJK_and_FullRight_glue:N

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_class_group_end:
        \__xeCJK_Default_and_FullRight_glue:N {#1}
        \__xeCJK_ulem_class_group_begin:
      }
      { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
  }
```

(*End definition for* `\__xeCJK_ulem_CJK_and_FullRight_glue:N`.)

__xeCJK_ulem_FullLeft_and_Default:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_Default:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
          {
            \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
            \__xeCJK_punct_bound_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
            \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
            \__xeCJK_punct_glue:NN \c__xeCJK_left_tl  \g__xeCJK_last_punct_tl
          }
```

```
          { \xeCJK_class_group_end: \UL@stop }
        \__xeCJK_ulem_skip_punct_end:
        \xeCJK_no_break:
        \UL@start
      }
      { \__xeCJK_ulem_FullLeft_and_Default: }
  }
```

(*End definition for* `\__xeCJK_ulem_FullLeft_and_Default:`.)

`\__xeCJK_ulem_FullLeft_and_CJK:`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_CJK:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_FullLeft_and_Default:
        \__xeCJK_ulem_class_group_begin:
      }
      { \__xeCJK_ulem_FullLeft_and_CJK: }
  }
```

(*End definition for* `\__xeCJK_ulem_FullLeft_and_CJK:`.)

`\__xeCJK_ulem_FullRight_and_Default:`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_Default:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \xeCJK_class_group_end:
        \UL@stop
        \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \__xeCJK_ulem_skip_punct_end:
        \UL@start
      }
      { \__xeCJK_ulem_FullRight_and_Default: }
  }
```

(*End definition for* `\__xeCJK_ulem_FullRight_and_Default:`.)

`\__xeCJK_ulem_FullRight_and_CJK:`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_CJK:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \xeCJK_class_group_end:
        \UL@stop
        \__xeCJK_punct_offset:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
        \__xeCJK_ulem_ccglue:
        \__xeCJK_ulem_skip_punct_end:
        \UL@start
        \__xeCJK_ulem_class_group_begin:
      }
      { \__xeCJK_ulem_FullRight_and_CJK: }
  }
```

(*End definition for* `\__xeCJK_ulem_FullRight_and_CJK:`.)

`\__xeCJK_ulem_punct_hskip:n`

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_hskip:n
  {
    \xeCJK_if_ulem_patch:TF
      { \__xeCJK_ulem_hskip:n }
      { \__xeCJK_ulem_punct_hskip:n }
  }
```

(*End definition for* `\__xeCJK_ulem_punct_hskip:n`.)

__xeCJK_ulem_punct_kern:n

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_kern:n
  {
    \xeCJK_if_ulem_patch:TF
      { \__xeCJK_ulem_hskip:n }
      { \__xeCJK_ulem_punct_kern:n }
  }
```

(*End definition for* __xeCJK_ulem_punct_kern:n.)

__xeCJK_ulem_punct_breakable_kern:n

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_punct_breakable_kern:n #1
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_class_group_end:
        \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start
        \__xeCJK_ulem_class_group_begin:
      }
      { \__xeCJK_ulem_punct_breakable_kern:n {#1} }
  }
```

(*End definition for* __xeCJK_ulem_punct_breakable_kern:n.)

__xeCJK_ulem_glue:n
__xeCJK_ulem_ccglue:
__xeCJK_ulem_hskip:n

在下划线状态下的分别代替 \CJKglue 等。

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_glue:n #1
  {
    \xeCJK_if_ulem_patch:TF
      { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
      { \skip_horizontal:n {#1} }
  }
\cs_new_protected_nopar:Npn \__xeCJK_ulem_ccglue:
  { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders }
\cs_new_protected_nopar:Npn \__xeCJK_ulem_hskip:n #1
  {
    \int_compare:nNnTF \tex_lastkern:D = \c_three
      { \skip_horizontal:n {#1} }
      { \skip_set:Nn \UL@skip {#1} \UL@leaders }
  }
```

(*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。

```
\RenewDocumentCommand \CJKunderdot { m }
  {
    \bool_if:NT \l__xeCJK_ulem_hook_used_bool
      { \UL@stop \__xeCJK_ulem_restore_CJK_and_Boundary: }
    \CJK@preUnderdot
    \__xeCJK_make_under_symbol:n { \CJK@underdotSkip }
    \cs_gset_eq:NN \__xeCJK_save_under_CJKsymbol:N \CJKsymbol
    \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
    \__xeCJK_restore_shipout_CJKsymbol:
    \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
    #1
    \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@stop }
    \cs_set_eq:NN \CJKsymbol \__xeCJK_save_under_CJKsymbol:N
    \tl_clear:N \l__xeCJK_fntef_shipout_tl
    \CJK@postUnderdot
    \bool_if:NT \l__xeCJK_ulem_hook_used_bool
      { \__xeCJK_ulem_restore_CJK_and_Boundary: \UL@start }
    \tex_ignorespaces:D
  }
\box_new:N \g__xeCJK_under_symbol_box
\cs_new_protected_nopar:Npn \__xeCJK_ulem_restore_CJK_and_Boundary:
  {
    \xeCJK_if_ulem_patch:TF
      {
        \xeCJK_swap_cs:NN
          \__xeCJK_CJK_and_Boundary_aux: \__xeCJK_ulem_CJK_and_Boundary_aux:
```

```
      }
      { }
  }
```

(*End definition for* `\CJKunderdot`.)

`\CJKunderanysymbol`

```
\RenewDocumentCommand \CJKunderanysymbol { m m m }
  {
    \group_begin:
    \hbox_set:Nn \CJK@underdotBox {#2}
    \__xeCJK_make_under_symbol:n {#1}
    \cs_set_eq:NN \__xeCJK_save_under_CJKsymbol:N \CJKsymbol
    \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
    \__xeCJK_restore_shipout_CJKsymbol:
    \tl_clear:N \l__xeCJK_underdot_shipout_tl
    #3
    \group_end:
    \tex_ignorespaces:D
  }
```

(*End definition for* `\CJKunderanysymbol`.)

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

```
\cs_new_protected:Npn \__xeCJK_restore_shipout_CJKsymbol:
  {
    \tl_set:Nn \l__xeCJK_fntef_shipout_tl
      {
        \l__xeCJK_underdot_shipout_tl
        \cs_set_eq:NN \CJKsymbol \__xeCJK_save_under_CJKsymbol:N
      }
  }
\tl_new:N \l__xeCJK_fntef_shipout_tl
\tl_new:N \l__xeCJK_underdot_shipout_tl
\tl_set:Nn \l__xeCJK_underdot_shipout_tl
  {
    \cs_set_eq:NN \CJKf@global \scan_stop:
    \int_zero:N \CJK@nest \CJK@postUnderdot
  }
\__xeCJK_add_to_shipout:n { \l__xeCJK_fntef_shipout_tl }
```

(*End definition for* `\__xeCJK_restore_shipout_CJKsymbol:`.)

`__xeCJK_make_under_symbol:n`

```
\cs_new_protected:Npn \__xeCJK_make_under_symbol:n #1
  {
    \hbox_set:Nn \l__xeCJK_tmp_box { ^^^^4e00 }
    \vbox_gset_to_ht:Nnn \g__xeCJK_under_symbol_box \c_zero_dim
      {
        \skip_vertical:n {#1}
        \hbox_to_zero:n
          {
            \tex_kern:D \box_wd:N \l__xeCJK_tmp_box
            \tex_hss:D \box_use:N \CJK@underdotBox \tex_hss:D
          }
        \tex_vss:D
      }
  }
```

(*End definition for* `\__xeCJK_make_under_symbol:n`.)

`\__xeCJK_under_CJKsymbol:N`

```
\cs_new_protected_nopar:Npn \__xeCJK_under_CJKsymbol:N #1
  {
    \hbox_overlap_right:n { \box_use:N \g__xeCJK_under_symbol_box }
    \xeCJK_no_break: \__xeCJK_save_under_CJKsymbol:N {#1}
  }
```

(*End definition for* `\__xeCJK_under_CJKsymbol:N`.)

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

```
\RenewDocumentEnvironment { CJKfilltwosides } { O { c } m }
  {
    \use:x { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
    \cs_set_eq:NN \CJKglue \tex_hfill:D
  }
  {
    \endminipage
    \ignorespacesafterend
  }
\NewEnviron { CJKfilltwosides* } [ 2 ] [ c ]
  {
    \cs_set_eq:NN \CJKglue \tex_hfill:D
    \tl_set:Nn \arraystretch { 1 }
    \token_if_dim_register:NT \extrarowheight
      { \dim_set_eq:NN \extrarowheight \c_zero_dim }
    \dim_compare:nNnTF {#2} > \c_zero_dim
      {
        \hbox_set:Nn \l__xeCJK_tmp_box
          {
            \tabular [#1] { @ { } c @ { } }
              \BODY
            \endtabular
          }
        \dim_compare:nNnTF {#2} > { \box_wd:N \l__xeCJK_tmp_box }
          {
            \tabular [#1] { @ { } p { \dim_eval:n {#2} } @ { } }
              \BODY
            \endtabular
          }
          { \hbox_unpack:N \l__xeCJK_tmp_box }
      }
      {
        \tabular [#1] { @ { } c @ { } }
          \BODY
        \endtabular
      }
  }
  [ \ignorespacesafterend ]
```

(*End definition for* CJKfilltwosides.)

⟨/fntef⟩

5.20 xeCJK-listings

仿照 luatexja 宏包中 lltjp-listings 的处理,支持 listings 宏包。

⟨*listings⟩

```
\DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
\ProcessOptions \scan_stop:
```

```
\RequirePackage { xeCJK }
\RequirePackage { listings }
```

```
\lst@AddToHook { Init } { \__xeCJK_listings_initial_hook: }
\lst@AddToHook { SelectCharTable } { \__xeCJK_listings_toks_hook: }
\lst@AddToHook { OutputBox }
  {
    \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
    \l__xeCJK_restore_listings_toks_tl
    \__xeCJK_listings_output_IVS:
  }
\lst@AddToHook { PreSet } { \bool_set_true:N \l__xeCJK_listings_env_bool }
```

__xeCJK_listings_initial_hook: 为使代码行号结果正确,需要在 \lst@numberstyle 中恢复 \XeTeXinterchartoks。在 listings 环境中换页时,对 \XeTeXinterchartoks 的修改会影响到页眉和页脚,需要在 \shipout 盒子中

恢复成正常定义。加入 \tex_noindent:D 是为了防止汉字出现在首行的时候可能会产生额外空行。

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_initial_hook:
  {
    \tex_noindent:D
    \bool_gset_false:N \g__xeCJK_listings_IVS_bool
    \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
    \__xeCJK_add_to_shipout:n { \l__xeCJK_restore_listings_toks_tl }
    \lst@ifbreaklines
      \cs_set_eq:NN \__xeCJK_listings_CJK_toks: \__xeCJK_listings_breaklines_toks:
    \fi:
  }
```

(*End definition for* __xeCJK_listings_initial_hook:.)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_toks_hook:
  {
    \tl_set:Nx \l__xeCJK_restore_listings_toks_tl
      {
        \__xeCJK_backup_inter_class_toks:nn { Boundary } { Default }
        \__xeCJK_backup_inter_class_toks:nn { Boundary } { CJK }
        \__xeCJK_backup_inter_class_toks:nn { Boundary } { IVS }
        \__xeCJK_backup_inter_class_toks:nn { Boundary } { FullLeft }
        \__xeCJK_backup_inter_class_toks:nn { Boundary } { FullRight }
      }
    \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
      {
        \tl_put_right:Nx \l__xeCJK_restore_listings_toks_tl
          { \__xeCJK_backup_inter_class_toks:nn { Boundary } { CJK/##1 } }
      }
    \xeCJK_inter_class_toks:nnn { Boundary } { Default }
      { \__xeCJK_listings_process_Default:N }
    \xeCJK_inter_class_toks:nnn { Boundary } { IVS }
      { \__xeCJK_listings_process_IVS:nN { \c_zero } }
    \__xeCJK_listings_CJK_toks_hook:
  }
\tl_new:N \l__xeCJK_restore_listings_toks_tl
\cs_new_nopar:Npn \__xeCJK_backup_inter_class_toks:nn #1#2
  {
    \xeCJK_inter_class_toks:nnn {#1} {#2}
      { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
  }
```

(*End definition for* __xeCJK_listings_toks_hook:.)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_CJK_toks_hook:
  {
    \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
      { \__xeCJK_listings_process_CJK:nN { \c_two } }
    \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
      { \__xeCJK_listings_process_CJK:nN { \c_two } }
    \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
      { \__xeCJK_listings_process_CJK:nN { \c_two } }
    \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
      {
        \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
          { \__xeCJK_listings_process_CJK:nN { \c_two } }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_listings_breaklines_toks:
  {
    \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
      { \__xeCJK_listings_process_breaklines_CJK:nN { \c_two } }
    \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
      { \__xeCJK_listings_process_FullLeft:nN { \c_two } }
    \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
```

```
      { \__xeCJK_listings_process_FullRight:nN { \c_two } }
    \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
      {
        \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
          { \__xeCJK_listings_process_breaklines_CJK:nN { \c_two } }
      }
  }
```

(*End definition for* `\__xeCJK_listings_CJK_toks_hook:` *and* `\__xeCJK_listings_breaklines_toks:`.)

`\__xeCJK_listings_process_Default:N`
`\__xeCJK_listings_process_CJK:nN`

对于 `\charcode` 大于 255 的字符，根据 `\catcode` 进行处理。

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_Default:N #1
  {
    \token_if_letter:NTF #1
      { \lst@ProcessLetter #1 }
      { \lst@ProcessOther  #1 }
  }
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_CJK:nN #1#2
  {
    \token_if_letter:NTF #2
      { \__xeCJK_listings_process_letter:nN {#1} #2 }
      { \__xeCJK_listings_process_other:nN  {#1} #2 }
  }
```

(*End definition for* `\__xeCJK_listings_process_Default:N` *and* `\__xeCJK_listings_process_CJK:nN`.)

`\__xeCJK_listings_append:nN`

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

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_append:nN #1#2
  {
    \int_add:Nn \lst@length { #1 - \c_one }
    \lst@Append #2
  }
```

(*End definition for* `\__xeCJK_listings_append:nN`.)

`\__xeCJK_listings_process_letter:nN`
`\__xeCJK_listings_process_other:nN`

在 `letter` 类中区分汉字和西文字母。

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_letter:nN
  {
    \lst@whitespacefalse
    \bool_if:NTF \l__xeCJK_listings_letter_bool
      { \lst@lettertrue }
      {
        \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
        \bool_set_true:N \l__xeCJK_listings_letter_bool
      }
    \__xeCJK_listings_append:nN
  }
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_other:nN #1#2
  {
    \lst@whitespacefalse
    \bool_if:NTF \l__xeCJK_listings_letter_bool
      {
        \lst@Output \lst@letterfalse
        \bool_set_false:N \l__xeCJK_listings_letter_bool
      }
      { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
    \cs_set_eq:NN \lst@lastother #2
    \__xeCJK_listings_append:nN {#1} #2
  }
```

(*End definition for* `\__xeCJK_listings_process_letter:nN` *and* `\__xeCJK_listings_process_other:nN`.)

`\__xeCJK_listings_process_breaklines_CJK:nN`
`\__xeCJK_listings_process_FullLeft:nN`
`\__xeCJK_listings_process_FullRight:nN`

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

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_breaklines_CJK:nN
  {
    \lst@whitespacefalse
    \bool_if:NTF \l__xeCJK_listings_letter_bool
```

```
      {
        \int_compare:nNnF \l__xeCJK_listings_flag_int = \c_two { \lst@Output }
        \lst@lettertrue
      }
      {
        \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
        \bool_set_true:N \l__xeCJK_listings_letter_bool
      }
    \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one
    \__xeCJK_listings_append:nN
  }
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_FullLeft:nN #1#2
  {
    \lst@whitespacefalse
    \bool_if:NTF \l__xeCJK_listings_letter_bool
      {
        \bool_if:nF
          {
            \int_compare_p:nNn \l__xeCJK_listings_flag_int = \c_two ||
            ( \int_compare_p:nNn \l__xeCJK_listings_flag_int = \c_three &&
              ! \l__xeCJK_punct_breakable_bool )
          }
          { \lst@Output }
        \lst@lettertrue
      }
      {
        \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
        \bool_set_true:N \l__xeCJK_listings_letter_bool
      }
    \int_set_eq:NN \l__xeCJK_listings_flag_int \c_two
    \__xeCJK_listings_append:nN {#1} #2
  }
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_FullRight:nN #1#2
  {
    \lst@whitespacefalse
    \bool_if:NTF \l__xeCJK_listings_letter_bool
      {
        \bool_if:nT
          {
            \int_compare_p:nNn \l__xeCJK_listings_flag_int < \c_two &&
            \__xeCJK_punct_if_long_p:N #2
          }
          { \lst@Output }
        \lst@lettertrue
      }
      {
        \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
        \bool_set_true:N \l__xeCJK_listings_letter_bool
      }
    \int_set_eq:NN \l__xeCJK_listings_flag_int \c_three
    \__xeCJK_listings_append:nN {#1} #2
  }
\int_new:N \l__xeCJK_listings_flag_int
```

(*End definition for* `\__xeCJK_listings_process_breaklines_CJK:nN`, `\__xeCJK_listings_process_FullLeft:nN`, *and* `\__xeCJK_listings_process_FullRight:nN`.)

`\lst@AppendLetter`
`\lst@AppendOther`

```
\cs_set_protected_nopar:Npn \lst@AppendLetter
  {
    \bool_if:NTF \l__xeCJK_listings_letter_bool
      {
        \lst@Output \lst@lettertrue
        \bool_set_false:N \l__xeCJK_listings_letter_bool
      }
      { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
    \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
    \lst@Append
  }
```

```
\cs_set_protected_nopar:Npn \lst@AppendOther
  {
    \bool_if:NTF \l__xeCJK_listings_letter_bool
      {
        \lst@Output \lst@letterfalse
        \bool_set_false:N \l__xeCJK_listings_letter_bool
      }
      { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
    \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
    \tex_futurelet:D \lst@lastother \lst@Append
  }
```

(*End definition for* `\lst@AppendLetter` *and* `\lst@AppendOther`.)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_process_IVS:nN
  {
    \reverse_if:N \lst@ifflexible
      \bool_gset_true:N \g__xeCJK_listings_IVS_bool
    \fi:
    \__xeCJK_listings_process_letter:nN
  }
```

(*End definition for* `\__xeCJK_listings_process_IVS:nN`.)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_listings_output_IVS:
  {
    \reverse_if:N \lst@ifflexible
      \bool_if:NT \g__xeCJK_listings_IVS_bool
        {
          \bool_gset_false:N \g__xeCJK_listings_IVS_bool
          \xeCJK_cs_clear:N \lst@FillOutputBox
          \cs_set_eq:NN \CJKglue \tex_hss:D
        }
    \fi:
  }
\bool_new:N \g__xeCJK_listings_IVS_bool
```

(*End definition for* `\__xeCJK_listings_output_IVS:`.)

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

```
\cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
  {
    \token_if_active:NTF #3
      { #1#3 }
      {
        \token_if_cs:NTF #3
          { #2#3 }
          {
            \int_compare:nNnTF { `#3 } > { \lst@ifec 255 \else: 127 \fi: }
              { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
              { #2#3 }
          }
      }
  }
\cs_set_eq:NN \lst@IfNextCharActive \__xeCJK_listings_peek_active_loop:TF
```

(*End definition for* `\__xeCJK_listings_peek_active_loop:TF`.)

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

```
\cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
  {
    \tl_set_rescan:Nnn \l__xeCJK_tmp_tl { } {#1}
    \__xeCJK_set_listings_escape:
```

```
    \tl_put_right:NV \lst@arg \l__xeCJK_tmp_tl
  }
\cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
\cs_new_protected_nopar:Npn \__xeCJK_listings_inline_group:w
  {
    \exp_after:wN \__xeCJK_listings_inline_group:n
    \exp_after:wN { \if_false: } \fi:
  }
\cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
\cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
  {
    \tl_set_rescan:Nnn \lst@arg { } {#1}
    \__xeCJK_set_listings_escape:
    \lst@InlineGJEnd
  }
```

(*End definition for* __xeCJK_listings_inside_convert:nw *and* __xeCJK_listings_inline_group:w.)

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

```
\group_begin:
\cs_set:Npn \__xeCJK_tmp:w #1
  {
    \group_end:
    \cs_new_protected:Npn \__xeCJK_set_listings_escape:
      { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
    \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
      { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
  }
\use:n
  {
    \char_set_catcode_active:N \\
    \__xeCJK_tmp:w
  }
  { \ }
```

(*End definition for* __xeCJK_set_listings_escape:.)

⟨/listings⟩

5.21 xunicode-addon

⟨*xunicode⟩

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

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

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

```
\pdftex_if_engine:T
  {
    \msg_new:nnnn { xunicode-addon } { cannot-use-pdftex }
      { This~package~requires~either~XeTeX~or~LuaTeX~to~function.}
      {
        You~must~change~your~typesetting~engine~to,~e.g.,\\
        "xelatex"~or~"lualatex"~instead~of~plain~"latex"~or~"pdflatex".
      }
    \msg_critical:nn { xunicode-addon } { cannot-use-pdftex }
  }
\RequirePackage { xparse }
```

宏包选项是编码的名字。

```
\clist_new:N \g__xunadd_encname_clist
\DeclareOption*
  { \clist_gput_left:NV \g__xunadd_encname_clist \CurrentOption }
\ProcessOptions \scan_stop:
\tl_if_exist:NT \UTFencname
  { \clist_gput_left:Nx \g__xunadd_encname_clist { \UTFencname } }
```

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

```
\@ifpackageloaded { xunicode } { }
  {
    \clist_get:NNF \g__xunadd_encname_clist \UTFencname
      {
        \xetex_if_engine:TF
          { \tl_set:Nn \UTFencname { EU1 } }
          { \tl_set:Nn \UTFencname { EU2 } }
        \clist_set_eq:NN \g__xunadd_encname_clist \UTFencname
      }
    \xetex_if_engine:TF
      { \RequirePackage { xunicode } }
      {
        \cs_set_eq:NN \__xunadd_tmp:w \XeTeXpicfile
        \cs_set_eq:NN \XeTeXpicfile \prg_do_nothing:
        \RequirePackage { xunicode }
        \cs_set_eq:NN \XeTeXpicfile \__xunadd_tmp:w
      }
  }
\AtEndOfPackage { \ReloadXunicode { \g__xunadd_encname_clist } }
```

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

```
\RenewDocumentCommand \ReloadXunicode { m }
  {
    \clist_set:Nx \l__xunadd_encname_clist {#1}
    \clist_remove_duplicates:N \l__xunadd_encname_clist
    \use:x
      {
        \bool_if:NT \l__kernel_expl_bool { \ExplSyntaxOff }
        \char_set_catcode_letter:n { 64 }
        \__xunadd_reload:N \exp_not:N \l__xunadd_encname_clist
        \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
        \bool_if:NT \l__kernel_expl_bool { \ExplSyntaxOn }
      }
  }
\cs_new_protected:Npn \__xunadd_reload:N #1
  {
    \cs_set_eq:NN \__xunadd_tmp:w \iftipaonetoken
    \cs_set_eq:NN \iftipaonetoken \scan_stop:
    \clist_map_inline:Nn #1
      {
        \cs_if_exist:cTF { T@ ##1 }
          {
            \tl_set:Nx \UTFencname {##1}
            \clist_gput_right:Nx \g__xunadd_encname_clist {##1}
            \file_input:n { xunicode.sty }
            \__xunadd_declare_textcircled:n { \UTFencname }
            \__xunadd_declare_math_symbols:n { \UTFencname }
          }
          { \msg_error:nnn { xunicode-addon } { encoding-unknown } {##1} }
      }
    \cs_set_eq:NN \iftipaonetoken \__xunadd_tmp:w
    \clist_gremove_duplicates:N \g__xunadd_encname_clist
  }
\clist_new:N \l__xunadd_encname_clist
\msg_new:nnnn { xunicode-addon } { encoding-unknown }
  { Encoding~scheme~"#1"~unknown. }
  {
    You~may~use \\\\
    \token_to_str:N \usepackage [ #1 , \encodingdefault ] {fontenc} \\\\
    before~xunicode-addon~or~xunicode.
  }
```

(*End definition for* \ReloadXunicode.)

__xunadd_declare_math_symbols: 几个特殊的数学符号对应的 Unicode 符号定义以 \text 打头的命令。

```
\cs_new_protected_nopar:Npn \__xunadd_declare_math_symbols:n #1
```

```
  {
    \__xunadd_declare_character:Nnn \textFinv   {#1} { x2132 }
    \__xunadd_declare_character:Nnn \textaleph  {#1} { x2135 }
    \__xunadd_declare_character:Nnn \textbeth   {#1} { x2136 }
    \__xunadd_declare_character:Nnn \textgimel  {#1} { x2137 }
    \__xunadd_declare_character:Nnn \textdaleth {#1} { x2138 }
    \__xunadd_declare_character:Nnn \textGame   {#1} { x2141 }
  }
```

(*End definition for* `\__xunadd_declare_math_symbols:.`)

`\DeclareUTFmathsymbols` 将文本符号定义为 `\protected` 宏后，为了与 hyperref 的书签功能兼容需要作一点额外处理。

```
\RenewDocumentCommand \DeclareUTFmathsymbols { m }
  {
    \bool_if:NT \l__xunadd_math_as_UTF_text_bool
      {
        \seq_map_inline:Nn \l__xunadd_math_as_UTF_text_seq
          { \__xunadd_declare_math_as_UTF_text:n {##1} }
        \bool_set_false:N \l__xunadd_math_as_UTF_text_bool
      }
  }
\seq_new:N \l__xunadd_math_as_UTF_text_seq
\seq_set_from_clist:Nn \l__xunadd_math_as_UTF_text_seq
  { hbar , Finv , aleph , beth , gimel , daleth , Game }
\bool_new:N \l__xunadd_math_as_UTF_text_bool
\RenewDocumentCommand \UseMathAsText { }
  {
    \math@s@text@true
    \bool_set_true:N \l__xunadd_math_as_UTF_text_bool
  }
\@onlypreamble \UseMathAsText
\cs_new_protected_nopar:Npn \__xunadd_declare_math_as_UTF_text:n #1
  {
    \cs_if_exist:cTF {#1}
      {
        \cs_new_eq:cc { keepmathUTF #1 } {#1}
        \cs_gset_protected_nopar:cpx {#1}
          {
            \exp_not:N \mode_if_math:TF
              { \exp_not:c { keepmathUTF #1 } }
              { \exp_not:c { text #1 } }
          }
        \tl_put_right:Nx \l__xunadd_hyperref_hook_tl
          { \cs_set_eq:NN \exp_not:c {#1} \exp_not:c { text #1 } }
      }
      { \cs_new_nopar:cpx {#1} { \exp_not:c { text #1 } } }
  }
\tl_new:N \l__xunadd_hyperref_hook_tl
\AtBeginDocument
  {
    \cs_if_free:NF \pdfstringdefDisableCommands
      { \pdfstringdefDisableCommands { \l__xunadd_hyperref_hook_tl } }
  }
```

(*End definition for* `\DeclareUTFmathsymbols.`)

`__xunadd_glyph_if_exist_p:n`
`__xunadd_glyph_if_exist:nTF` 判断字符在当前字体中是否存在。

```
\prg_new_conditional:Npnn \__xunadd_glyph_if_exist:n #1 { p , T , F , TF }
  {
    \etex_iffontchar:D \tex_font:D \etex_numexpr:D #1 \scan_stop:
      \prg_return_true: \else: \prg_return_false: \fi:
  }
```

(*End definition for* `\__xunadd_glyph_if_exist:n.`)

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

```
\RenewDocumentCommand \UndeclareUTFcharacter { O { \UTFencname } m m }
  {
    \__xunadd_if_csname:nTF {#3}
      { \UndeclareTextCommand {#3} }
```

```
      { \exp_args:Nc \UndeclareTextCommand { \tl_to_str:n {#3} } }
      {#1}
  }
```

(*End definition for* *UndeclareUTFcharacter*.)

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

```
\RenewDocumentCommand \UndeclareUTFcomposite { O { \UTFencname } m m m }
  {
    \__xunadd_if_csname:nTF {#3}
      { \__xunadd_undeclare_composite:Nnnn #3 }
      { \exp_args:Nc \__xunadd_undeclare_composite:Nnnn { \tl_to_str:n {#3} } }
      {#1} {#4} {#2}
  }
\cs_new_protected:Npn \__xunadd_undeclare_composite:Nnnn #1#2#3#4
  { \cs_undefine:c { \token_to_str:c {#2} \token_to_str:N #1 - \tl_to_str:n {#3} } }
```

(*End definition for* *UndeclareUTFcomposite*.)

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

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

```
\prg_new_conditional:Npnn \__xunadd_if_csname:n #1 { TF }
  {
    \tl_if_single_token:nTF {#1}
      {
        \if_predicate:w
          \bool_if_p:n { \token_if_cs_p:N #1 || \token_if_active_p:N #1 }
          \prg_return_true: \else: \prg_return_false: \fi:
      }
      { \prg_return_false: }
  }
```

(*End definition for* *__xunadd_if_csname:nTF*.)

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

```
\RenewDocumentCommand \DeclareUTFcharacter { O { \UTFencname } m m }
  {
    \str_if_eq:nnTF {#3} { \hbar }
      { \__xunadd_restore_hbar: }
      {
        \__xunadd_if_csname:nTF {#3}
          { \__xunadd_declare_character:Nnn #3 }
          { \__xunadd_declare_character:cnn { \tl_to_str:n {#3} } }
        {#1} {#2}
      }
  }
```

(*End definition for* *DeclareUTFcharacter*.)

__xunadd_restore_hbar: 恢复 \hbar 为原本定义。

```
\cs_new_protected_nopar:Npn \__xunadd_restore_hbar:
  {
    \cs_if_free:cF { ? - \token_to_str:N \hbar }
      { \__xunadd_restore_hbar:c { ? - \token_to_str:N \hbar } }
  }
\cs_new_protected_nopar:Npn \__xunadd_restore_hbar:N #1
  {
    \cs_gset_eq:NN \hbar #1
    \cs_undefine:N #1
  }
\cs_generate_variant:Nn \__xunadd_restore_hbar:N { c }
```

(*End definition for* *__xunadd_restore_hbar:*.)

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

```
\cs_new_protected:Npn \__xunadd_declare_character:Nnn #1#2#3
  {
    \__xunadd_provide_text_command_default:N #1
    \group_begin:
    \char_set_lccode:nn { `0 } { \__xunadd_check_slot:n {#3} }
```

```
    \tl_to_lowercase:n
      {
        \group_end:
        \__xunadd_declare_character:NNxn 0
      }
      #1 { \token_to_str:N #1 } {#2}
  }
\cs_generate_variant:Nn \__xunadd_declare_character:Nnn { c }
```

(*End definition for* __xunadd_declare_character:Nnn.)

__xunadd_provide_text_command_default:N

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

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

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

```
\cs_new_protected:Npn \__xunadd_provide_text_command_default:N #1
  {
    \bool_if:nF
      {
        \cs_if_exist_p:c { ?   \token_to_str:N #1 } ||
        \cs_if_free_p:c  { ? - \token_to_str:N #1 }
      }
      { \exp_args:NNv \ProvideTextCommandDefault #1 { ? - \token_to_str:N #1 } }
  }
```

(*End definition for* __xunadd_provide_text_command_default:N.)

__xunadd_declare_character:NNnn

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

```
\cs_new_protected:Npn \__xunadd_declare_character:NNnn #1#2#3#4
  { \DeclareTextCommand #2 {#4} { \__xunadd_text_character:nN {#3} {#1} } }
\cs_new_protected:Npn \__xunadd_text_character:nN #1#2
  {
    \__xunadd_begin_hook:nn {#1} {#2}
    \__xunadd_glyph_if_exist:nTF { `#2 }
      {#2} { \cs_if_exist_use:cF { ? #1 } {#2} }
    \__xunadd_end_hook:nn {#1} {#2}
  }
\cs_generate_variant:Nn \__xunadd_declare_character:NNnn { NNx }
```

(*End definition for* __xunadd_declare_character:NNnn.)

__xunadd_check_slot:n

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

```
\cs_new_nopar:Npn \__xunadd_check_slot:n #1
  {
    \int_eval:n
      {
        \tl_if_head_eq_charcode:nNTF {#1} x
          { " \use_none:n #1 } {#1}
      }
  }
```

(*End definition for* __xunadd_check_slot:n.)

\DeclareUTFcomposite

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

```
\RenewDocumentCommand \DeclareUTFcomposite { O { \UTFencname } m m m }
  {
    \__xunadd_if_csname:nTF {#3}
      { \__xunadd_declare_composite:Nnnn #3 }
```

```
      { \__xunadd_declare_composite:cnnn { \tl_to_str:n {#3} } }
      {#1} {#4} {#2}
  }
```

(*End definition for* *DeclareUTFcomposite*.)

__xunadd_declare_composite:Nnnn

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

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

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

```
\cs_new_protected:Npn \__xunadd_declare_composite:Nnnn #1#2#3#4
  {
    \tex_afterassignment:D \use_none_delimit_by_q_stop:w
    \__xunadd_chardef:cn
      { \token_to_str:c {#2} \token_to_str:N #1 - \tl_to_str:n {#3} }
      { \__xunadd_check_slot:n {#4} }
    \q_stop
  }
\cs_new_protected:Npn \__xunadd_chardef:Nn #1#2
  { \tex_chardef:D #1 = \etex_numexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \__xunadd_chardef:Nn { c }
\cs_generate_variant:Nn \__xunadd_declare_composite:Nnnn { c }
```

(*End definition for* *__xunadd_declare_composite:Nnnn*.)

\tonebar

使得 \tonebar{25} 和 \tonebar{52} 生效。

```
\AtEndOfPackage
  {
    \DeclareUTFCompositeCommand \tonebar { 25 } { \tonebar { 2 } \tonebar { 5 } }
    \DeclareUTFCompositeCommand \tonebar { 52 } { \tonebar { 5 } \tonebar { 2 } }
  }
```

(*End definition for* *tonebar*.)

\DeclareUTFCompositeCommand

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

```
\NewDocumentCommand \DeclareUTFCompositeCommand { O { \UTFencname } m m m }
  {
    \__xunadd_if_csname:nTF {#2}
      { \__xunadd_declare_composite_command:Nnnn #2 }
      { \__xunadd_declare_composite_command:cnnn { \tl_to_str:n {#2} } }
      {#1} {#3} {#4}
  }
```

(*End definition for* *DeclareUTFCompositeCommand*.)

__xunadd_declare_composite_command:Nnnn

```
\cs_new_protected:Npn \__xunadd_declare_composite_command:Nnnn #1#2#3#4
  {
    \cs_set_protected:cpn
      { \token_to_str:c {#2} \token_to_str:N #1 - \tl_to_str:n {#3} }
      {#4}
  }
\cs_generate_variant:Nn \__xunadd_declare_composite_command:Nnnn { c }
```

(*End definition for* *__xunadd_declare_composite_command:Nnnn*.)

\DeclareEncodedCompositeCharacter

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

```
\RenewDocumentCommand \DeclareEncodedCompositeCharacter { m m m m }
  {
    \__xunadd_if_csname:nTF {#2}
      { \__xunadd_declare_accent:Nnnn #2 }
      { \__xunadd_declare_accent:cnnn { \tl_to_str:n {#2} } }
      {#1} {#3} {#4}
  }
```

(*End definition for* *DeclareEncodedCompositeCharacter*.)

\DeclareEncodedCompositeAccents

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

```
\RenewDocumentCommand \DeclareEncodedCompositeAccents { m m m m }
  {
    \__xunadd_if_csname:nTF {#2}
      { \__xunadd_declare_accent:Nnnn #2 }
      { \__xunadd_declare_accent:cnnn { \tl_to_str:n {#2} } }
      {#1} {#4} {#3}
  }
```

(*End definition for* \DeclareEncodedCompositeAccents.)

__xunadd_declare_accent:Nnnn

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

```
\cs_new_protected:Npn \__xunadd_declare_accent:Nnnn #1#2#3#4
  {
    \group_begin:
    \char_set_lccode:nn { `3 } { "#3 }
    \char_set_lccode:nn { `4 } { \tl_if_blank:nTF {#4} { "#3 } { "#4 } }
    \tl_to_lowercase:n
      {
        \group_end:
        \__xunadd_declare_accent:NNNxxn 3 4
      }
      #1 { \token_to_str:c {#2} } { \token_to_str:N #1 } {#2}
  }
\cs_new_protected:Npn \__xunadd_declare_accent:NNNnnn #1#2#3#4#5#6
  {
    \DeclareTextCommand #3 {#6}
      { \__xunadd_text_composite:nnNNn {#4} {#5} {#1} {#2} }
  }
\cs_generate_variant:Nn \__xunadd_declare_accent:Nnnn { c }
\cs_generate_variant:Nn \__xunadd_declare_accent:NNNnnn { NNNxx }
```

(*End definition for* __xunadd_declare_accent:Nnnn.)

__xunadd_text_composite:nnNNn

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

```
\cs_new_protected:Npn \__xunadd_text_composite:nnNNn #1#2#3#4#5
  {
    \__xunadd_begin_hook:nn {#2} {#5}
    \cs_if_exist:cTF { #1#2 - \tl_to_str:n {#5} }
      {
        \__xunadd_text_composite_aux:cnNNn
          { #1#2 - \tl_to_str:n {#5} } {#2} {#3} {#4} {#5}
      }
      { \__xunadd_add_accent:nnNN {#5} {#2} {#3} {#4} }
    \__xunadd_end_hook:nn {#2} {#5}
  }
\cs_new_protected:Npn \__xunadd_text_composite_aux:NnNNn #1#2#3#4#5
  {
    \token_if_chardef:NTF #1
      {
        \__xunadd_glyph_if_exist:nTF {#1}
          {#1} { \__xunadd_add_accent:nnNN {#5} {#2} {#3} {#4} }
      }
      {#1}
  }
\cs_generate_variant:Nn \__xunadd_text_composite_aux:NnNNn { c }
```

(*End definition for* __xunadd_text_composite:nnNNn.)

__xunadd_add_accent:nnNN

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

[11] http://tug.org/pipermail/xetex/2013-July/024579.html

```
\cs_new_protected:Npn \__xunadd_add_accent:nnNN #1#2#3#4
  {
    \tl_if_blank:nTF {#1}
      {
        \__xunadd_glyph_if_exist:nTF { `#4 }
          {#4}
          { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
      }
      {
        \bool_if:NTF \l__xunadd_add_circle_bool
          {
            \bool_set_false:N \l__xunadd_add_circle_bool
            \__xunadd_glyph_if_exist:nTF { `#4 }
              { \__xunadd_add_circle:nn {#4} {#1} }
          }
          {
            \__xunadd_glyph_if_exist:nTF { `#3 }
              { #1#3 }
              {
                \__xunadd_glyph_if_exist:nTF { `#4 }
                  { \add@accent { `#4 } {#1} }
              }
          }
          { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
      }
  }
```

(*End definition for* __xunadd_add_accent:nnNN.)

__xunadd_add_circle:nn 对圆圈中的数字或者字母适当缩小，以适合圆圈的大小。只有字体中存在 U+25EF 时，才使用这里的设置，否则还还是 LaTeX 中的设置。

```
\cs_new_protected:Npn \__xunadd_add_circle:nn #1#2
  {
    \hcoffin_set:Nn \l__xunadd_circle_coffin {#1}
    \hcoffin_set:Nn \l__xunadd_tmp_coffin {#2}
    \dim_set:Nn \l__xunadd_circle_width_dim
      { \fp_use:N \l__xunadd_circle_ratio_fp \coffin_wd:N \l__xunadd_circle_coffin }
    \coffin_resize:Nnn \l__xunadd_tmp_coffin
      { \l__xunadd_circle_width_dim }
      {
        ( \coffin_ht:N \l__xunadd_tmp_coffin + \coffin_dp:N \l__xunadd_tmp_coffin )
        * \tex_number:D \l__xunadd_circle_width_dim
        / \tex_number:D \coffin_wd:N \l__xunadd_tmp_coffin
      }
    \coffin_attach:NnnNnnnn
      \l__xunadd_circle_coffin { hc } { vc }
      \l__xunadd_tmp_coffin    { hc } { vc }
      { \c_zero_dim } { \c_zero_dim }
    \coffin_typeset:Nnnnn \l__xunadd_circle_coffin
      { H } { l } { \c_zero_dim } { \c_zero_dim }
  }
\dim_new:N \l__xunadd_circle_width_dim
\coffin_new:N \l__xunadd_tmp_coffin
\coffin_new:N \l__xunadd_circle_coffin
```

(*End definition for* __xunadd_add_circle:nn.)

\settextcircledratio 设置圆圈中文字的宽度与圆圈宽度的比例，预设为 0.7。

```
\NewDocumentCommand \settextcircledratio { m }
  { \fp_set:Nn \l__xunadd_circle_ratio_fp {#1} }
\fp_new:N \l__xunadd_circle_ratio_fp
\settextcircledratio { 0.7 }
```

(*End definition for* \settextcircledratio.)

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

```
\NewDocumentCommand \AtBeginUTFCommand { s O { } +m }
  {
    \tl_if_blank:nTF {#2}
```

```
      {
        \IfBooleanTF {#1}
          { \tl_set:Nn \l__xunadd_begin_hook_tl {#3} }
          { \tl_put_right:Nn \l__xunadd_begin_hook_tl {#3} }
      }
      { \__xunadd_set_cmd_hook:nnn { begin } {#2} {#3} }
  }
\NewDocumentCommand \AtEndUTFCommand { s O { } +m }
  {
    \tl_if_blank:nTF {#2}
      {
        \IfBooleanTF {#1}
          { \tl_set:Nn \l__xunadd_end_hook_tl {#3} }
          { \tl_put_right:Nn \l__xunadd_end_hook_tl {#3} }
      }
      { \__xunadd_set_cmd_hook:nnn { end } {#2} {#3} }
  }
\tl_new:N \l__xunadd_begin_hook_tl
\tl_new:N \l__xunadd_end_hook_tl
```

(*End definition for* `\AtBeginUTFCommand` *and* `\AtEndUTFCommand`.)

`\__xunadd_set_cmd_hook:nnn`

```
\cs_new_protected:Npn \__xunadd_set_cmd_hook:nnn #1#2#3
  {
    \cs_set_protected:cpn
      {
        \tl_if_single:nTF {#2}
          { \use:c { __xunadd_#1_csname:n } { \token_to_str:N #2 } }
          { \__xunadd_set_cmd_hook_aux:Nnwn #2 \q_stop {#1} }
      } ##1
      {#3}
  }
\cs_new:Npn \__xunadd_set_cmd_hook_aux:Nnwn #1#2 \q_stop #3
  { \use:c { __xunadd_#3_csname:n } { \token_to_str:N #1 - \tl_to_str:n {#2} } }
\cs_new_nopar:Npn \__xunadd_begin_csname:n #1 { __xunadd_begin_#1_hook:n }
\cs_new_nopar:Npn \__xunadd_end_csname:n #1   { __xunadd_end_#1_hook:n }
```

(*End definition for* `\__xunadd_set_cmd_hook:nnn`.)

`\__xunadd_begin_hook:nn`
`\__xunadd_end_hook:nn`

```
\cs_new_protected:Npn \__xunadd_begin_hook:nn #1#2
  {
    \tl_use:N \l__xunadd_begin_hook_tl
    \cs_if_exist_use:cF { \__xunadd_begin_csname:n { #1 - \tl_to_str:n {#2} } }
      { \cs_if_exist_use:cF { \__xunadd_begin_csname:n {#1} } { \use_none:n } }
      {#2}
  }
\cs_new_protected:Npn \__xunadd_end_hook:nn #1#2
  {
    \cs_if_exist_use:cF { \__xunadd_end_csname:n { #1 - \tl_to_str:n {#2} } }
      { \cs_if_exist_use:cF { \__xunadd_end_csname:n {#1} } { \use_none:n } }
      {#2}
    \tl_use:N \l__xunadd_end_hook_tl
  }
```

(*End definition for* `\__xunadd_begin_hook:nn` *and* `\__xunadd_end_hook:nn`.)

`\__xunadd_declare_textcircled:n` 启用 xunicode 中的带圈数字和字母设置。

```
\AtBeginUTFCommand [ \textcircled ]
  { \bool_set_true:N  \l__xunadd_add_circle_bool }
\bool_new:N \l__xunadd_add_circle_bool
\cs_new_protected_nopar:Npn \__xunadd_declare_textcircled:n #1
  {
    \__xunadd_declare_character:Nnn \textbigcircle {#1} { x25EF }
    \__xunadd_declare_accent:Nnnn \textcircled {#1} { 20DD } { 25EF }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} {  0 } { x24EA }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} {  1 } { x2460 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} {  2 } { x2461 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} {  3 } { x2462 }
```

```
\__xunadd_declare_composite:Nnnn \textcircled {#1} {  4 } { x2463 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} {  5 } { x2464 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} {  6 } { x2465 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} {  7 } { x2466 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} {  8 } { x2467 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} {  9 } { x2468 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 10 } { x2469 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 11 } { x246A }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 12 } { x246B }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 13 } { x246C }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 14 } { x246D }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 15 } { x246E }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 16 } { x246F }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 17 } { x2470 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 18 } { x2471 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 19 } { x2472 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 20 } { x2473 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 21 } { x3251 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 22 } { x3252 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 23 } { x3253 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 24 } { x3254 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 25 } { x3255 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 26 } { x3256 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 27 } { x3257 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 28 } { x3258 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 29 } { x3259 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 30 } { x325A }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 31 } { x325B }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 32 } { x325C }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 33 } { x325D }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 34 } { x325E }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 35 } { x325F }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 36 } { x32B1 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 37 } { x32B2 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 38 } { x32B3 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 39 } { x32B4 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 40 } { x32B5 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 41 } { x32B6 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 42 } { x32B7 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 43 } { x32B8 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 44 } { x32B9 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 45 } { x32BA }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 46 } { x32BB }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 47 } { x32BC }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 48 } { x32BD }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 49 } { x32BE }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { 50 } { x32BF }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { A } { x24B6 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { B } { x24B7 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { C } { x24B8 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { D } { x24B9 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { E } { x24BA }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { F } { x24BB }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { G } { x24BC }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { H } { x24BD }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { I } { x24BE }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { J } { x24BF }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { K } { x24C0 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { L } { x24C1 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { M } { x24C2 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { N } { x24C3 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { O } { x24C4 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { P } { x24C5 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { Q } { x24C6 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { R } { x24C7 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { S } { x24C8 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { T } { x24C9 }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { U } { x24CA }
\__xunadd_declare_composite:Nnnn \textcircled {#1} { V } { x24CB }
```

```
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { W } { x24CC }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { X } { x24CD }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { Y } { x24CE }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { Z } { x24CF }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { a } { x24D0 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { b } { x24D1 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { c } { x24D2 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { d } { x24D3 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { e } { x24D4 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { f } { x24D5 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { g } { x24D6 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { h } { x24D7 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { i } { x24D8 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { j } { x24D9 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { k } { x24DA }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { l } { x24DB }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { m } { x24DC }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { n } { x24DD }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { o } { x24DE }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { p } { x24DF }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { q } { x24E0 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { r } { x24E1 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { s } { x24E2 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { t } { x24E3 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { u } { x24E4 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { v } { x24E5 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { w } { x24E6 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { x } { x24E7 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { y } { x24E8 }
    \__xunadd_declare_composite:Nnnn \textcircled {#1} { z } { x24E9 }
  }
```

(*End definition for* `\__xunadd_declare_textcircled:n`.)

⟨/xunicode⟩

5.22 xeCJK.cfg

⟨*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。

⟨/config⟩

版本历史

v3.1.0

General: 使用 xtemplate 宏包的机制来组织标点符号的处理。 47

删除多余的 default-itcorr 结点。 30

取消 \cprotect 的外部宏限制。 83

改用 indentfirst 宏包处理缩进的问题。 76

放弃使用放缩字体大小的方式，而只采用调整间距的方式与西文等宽字体对齐。并且只适用于与抄录环境下。 . . 70

放弃对 \outer 宏的特殊处理。 1

__xeCJK_switch_font:nn: 改进定义，加快切换速度。 . . 63

\c_xeCJK_space_skip_tl: 字间空格考虑 \spaceskip 不为零的情况。 16

LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。 75

\xeCJK@fix@penalty: 采用通过不修改原语 \/ 的方式对修复倾斜校正。 80

\xeCJK_fallback_loop:Nn: 调整备用字体的循环方式。 . . 56

\xeCJK_glyph_if_exist:N: 改进 fontspec 宏包中定义的 \font_glyph_if_exist:NnTF。 15

\xeCJK_hook_for_ulem:: 简化对 ulem 宏包的兼容补丁。 . . 84

\xeCJK_visible_space_fallback:: 调整 fontspec 的后备可视空格符号，以便于使用时对齐。 74

\xeCJKVerbAddon: 新增 \xeCJKVerbAddon 用于抄录环境中的间距调整。 71

v3.1.1

General: 不再依赖 xpatch 宏包。 1

增加 NewLineCS 和 EnvCS 选项。 42

增加小宏包 xeCJKfntef，用于处理下划线的问题。 84

对于与 xltxtra 的冲突给出错误警告。 80

__xeCJK_check_single_space:NN: CheckSingle 支持段末“汉字 + 汉字 + 空格 + 汉字/标点”的形式。 41

__xeCJK_set_char_class_eq:nn: 交换参数的顺序。 . . . 26

__xeCJK_set_verb_exspace:: 调整间距的计算方法。 . . 73

\fontfamily: 修改主要 CJK 字体族的自动更新方式。 80

CheckFullRight: 处理全角右标点之后的断行问题。 38

PlainEquation: 增加 PlainEquation 选项。 43

InlineEnv: 改变行内环境的设置方式，从而使用 \str_case_x:nnn 代替原来的 \clist_if_in:NnTF 来判断是否是行内环境。 43

\xeCJK_check_single:NNw: 改进定义，减少使用 peek 函数的次数。 41

\xeCJK_hook_for_ulem:: 完全处理下划线里的标点符号的有关问题。 84

\xeCJK_peek_catcode_ignore_spaces:NTF: 新增有省略空格标识的 peek 函数。 17

\xeCJK_save_class:nn: 使用 \xeCJK_save_class:nn 保存 XeTeX 预定义的字符类别。 19

\xeCJK_set_char_class:nnn: 在文档中设置字符类别时不重复设置 \catcode。 25

\xeCJKnobreak: 增加 \nobreak 的 xeCJK 版本。 40

v3.1.2

General: 修正重定义 \CJKfamilydefault 无效的问题，恢复容错能力。 68

解决在下划线状态下使用 \makebox 时的错误。 86

__xeCJK_check_single_space:NN: 使用 \xeCJK_if_CJK_class:NTF 来代替 \int_case:nnn 判断是否是 CJK 字符类。 41

__xeCJK_family_unknown_warning:n: 在没有定义任何 CJK 字体的情况下，不再重复给出字体没有定义的警告。 65

\fontfamily: 不将参数完全展开。 80

\nobreakspace: 修正非 \UTFencname 编码下面 xunicode 重定义的 \nobreakspace 会失效的问题。 79

v3.2.0

General: 增加 IVS 字符类用于处理异体字选择符。 20

增加 Verb 选项。 70

__xeCJK_Boundary_and_FullLeft_glue:N: 当全角左标点前面是 hlist、none、glue 和 penalty 等节点时，压缩其左空白。 35

\c_xeCJK_space_skip_tl: 字间空格考虑到 \spacefactor 和 \xspaceskip 的情况。 16

\l_xeCJK_family_tl: 不将其初始化为 \CJKfamilydefault。 65

\setCJKmonofont: 定义中加入 \normalfont。 65

\xeCJK_FullLeft_and_Default:: 修正 xeCJK 使西文在部分情况下无法断词的问题。 34

v3.2.1

General: 调整 Verb 选项：在命令 \verb 里使用时，不破坏标点禁则，增加值 env+。 70

v3.2.2

General: 修正某些重音不能正确显示的问题。 1

增加小宏包 xeCJK-listings，用于支持 listings 宏包。 92

__xeCJK_ulem_CJK_and_FullRight_glue:N: 修正下划线不能跳过全角右标点的问题。 88

v3.2.3

General: 不再改变 CJK 字符类的 \catcode。 25

完善对 listings 宏包的支持。 92

提供四个 TECkit 映射文件用于句号转换和简繁互换。 . . 1

根据 XeTeX 的脚本重新整理全角标点符号。 20

解决 CheckSingle 选项与 tablists 宏包的冲突。 42

__xeCJK_listings_initial_hook:: 解决 listings 坏境中代码行号输出不正确的问题，并解决在其中跨页时对页眉和页脚的影响。 92

__xeCJK_listings_process_CJK:nN: 在 listings 坏境中对 \charcode 大于 255 的字符根据其 \catcode 区分 letter 和 other。 94

__xeCJK_restore_shipout_CJKsymbol:: 解决 \CJKunderdot 跨页使用时影响到页眉页脚的问题。 91

__xeCJK_ulem_FullLeft_and_CJK:: 修正全角左标点后下划线与 \CJKunderdot 连用时结果不正常的问题。 . . . 89

\xeCJKVerbAddon: 新增 \xeCJKOffVerbAddon 用于局部取消 \xeCJKOffVerbAddon 的影响；并解决跨页使用时影响到页眉页脚的问题。 71

v3.2.4

General: 不再使用 CJKnumber 选项，可以在 xeCJK 之后直接使用 CJKnumb 宏包得到中文数字。 83

使 listings 的 breaklines 选项对 CJK 字符类可用，并保持标点符号的禁则。 94

使用 AllowBreakBetweenPuncts 时，相应标点符号仍能与边界对齐。 38

修正 xeCJKfntef 与 natbib 等的冲突。 84

内部调整分区字体的设置方法。 58

尽量移除用作判断标志的 \kern。 30

改进获取分区字体属性的办法。 58

解决使用 CheckSingle 时，某些 \CJKglue 不能被正确加入的问题。 42

遵循 LaTeX3 变量需要预先声明的原则。 1

__xeCJK_Boundary_and_FullLeft_glue:N: 细化边界与全角左标点之间是否压缩空白的判断。 35

__xeCJK_set_verb_exspace:: 当计算得出的间距为负时，缩小 CJK 字体。 73

\addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。 66
CJKfilltwosides: 改用 minipage 和 LaTeX 表格(tabular)来实现。 92
\xeCJK_fallback_loop:Nn: 使 \CJKfamilydefault 的 FallBack 设置全局可用。 56
\xeCJK_tl_remove_outer_braces:n: 去掉外层分组括号时,移除空格,避免死循环。 15
\xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。 18
v3.2.5
General: 修正 CJK 和 NormalSpace 字符类之间因为边界造成的间距不正确的问题。 29
增加小宏包 xunicode-addon, 为 xunicode 提供判断字符是否存在的功能。 97
\]: 解决 fixltx2e 和 amsthm 的冲突。 79
__xeCJK_Boundary_and_FullLeft_glue:N: 增加对 enumitem 宏包修改的 \item 的判断。 35
细化全角左标点是否位于段首的判断。 35
__xeCJK_math_robust:N: 解决汉字后紧跟 \(...\) 形式的行内数学公式时,不能加入间距的问题。 78
Verb: 微调定义。 70
\nobreakspace: 恢复 \nobreakspace 的原始定义。 79
\xeCJK_visible_space:: 可视空格考虑传统 TeX 字体的情况。 74
\xeCJKVerbAddon: 禁止自动换行,与西文一致。 71
v3.2.6
General: AutoFakeBold 和 AutoFakeSlant 选项直接使用 fontspec 的设置,修正不能调用相应实际字体的问题。 . . 59
case 类函数的用法与 LaTeX3 同步。 1
__xeCJK_math_robust:N: 考虑 \math 和 \ensuremath。 78
考虑 ulem 对 \MakeRobust 的不当定义。 78
\AtEndUTFCommand: 可以指定特定符号命令使用的钩子。 104
\mathrm: 为 \mathrm 减少一个可能的数学字体族。 78
\xeCJK_CJK_and_Boundary:w: 更好的处理边界是 \relax 的情况。 32
\xeCJK_set_mathfont:: 设置粗体时先检查对应字体是否存在。 69
v3.2.7
General: 使用 everypage 往 \shipout 盒子里加钩子。 14
修正 unicode-letters.tex 中谚文符号 \catcode 不准的问题。 25
处理 AllowBreakBetweenPuncts 与 xeCJKfntef 的兼容问题。 38
实现自定义行首/尾标点符号宽度功能。 45
标点宽度设置禁用比例选项的值改为 nan。 51
__xeCJK_punct_glue:NN: 标点符号左/右空白的伸展值不超过原始边界,收缩值不小于另一侧边界。 34
\Url@MathSetup: 使通过 \UrlFont 等命令设置的 CJK 字体生效。 78
\xeCJK_check_single:NNw: 与 \CJKspace 兼容。 41
\xeCJK_set_mathfont:: 将 CJK 字符的数学归类由 7 改为 0,解决汉字路径的问题。 69
v3.2.8
General: 启用 xunicode 中的带圈数字和字母设置。 105
__xeCJK_nobreak_skip:: 禁止在 \verb 中断行。 70
\DeclareUTFmathsymbols: 修正 \UseMathAsText 的功能,恢复 \hbar 和增加以 text 打头的文本符号命令。 99
\xeCJKVerbAddon: 增加是否是等宽字体的判断。 71

代码索引

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

Symbols

\(. 3447
\) 3447
\@@italiccorr 3555, 3563, 3569, 3575, 3578, 3589
\@empty 4429
\@ifpackagelater 46, 49, 3516, 3532
\@ifpackageloaded 4482
\@onefilewithoptions 3740
\@onlypreamble 1458, 1576, 1580, 2166, 2173, 2761, 2762, 2763, 2764, 2765, 2802, 4568
\@pkgextension 16
\[. 3355, 3496
\\ 5, 6, 7, 19, 24, 40, 41, 42, 43, 355, 356, 1494, 1548, 1549, 2149, 2150, 2151, 2163, 2164, 2256, 2721, 2854, 2936, 2937, 2938, 2939, 2944, 3224, 3226, 3332, 3377, 3378, 3488, 3510, 3511, 4459, 4470, 4537, 4538
\] 3496
__xeCJK_Boundary_and_FullLeft_glue:N 1025, 1034, 1034, 3817
__xeCJK_CJK_and_Boundary_aux: . 862, 866, 870, 3812, 4106
__xeCJK_CJK_and_Boundary_relax:N 861, 868
__xeCJK_CJK_and_FullLeft_glue:N 1011, 1016, 3815
__xeCJK_CJK_and_FullRight_glue:N 1112, 1116, 1116, 1130, 3816
__xeCJK_CJK_class_tl:c 472
__xeCJK_CJK_class_tl:n 306, 309, 311
__xeCJK_Default_and_FullLeft_glue:N . . . 994, 1003, 3813
__xeCJK_Default_and_FullRight_glue:N 1091, 1102, 1116, 1130, 3814, 3983
__xeCJK_add_special_punct:nn 1598, 1601, 1604, 1629
__xeCJK_add_sub_class_features:n 2824, 2831, 2857, 2857
__xeCJK_add_to_shipout:n . . . 108, 3080, 3142, 4138, 4216
__xeCJK_after_end_preamble:n 71, 78, 3430, 3591, 3659, 3672
__xeCJK_after_preamble:n 71, 76, 3016, 3639
__xeCJK_at_end_preamble:n . 71, 74, 2891, 3528, 3694, 3707
__xeCJK_aux_r:n 3642, 3650, 3651
__xeCJK_backup_inter_class_toks:nn 4225, 4226, 4227, 4228, 4229, 4234, 4243
__xeCJK_block_select_font:n 2622, 2626, 2626
__xeCJK_calc_kerning_margin:NN 1980, 2015, 2015
__xeCJK_calc_kerning_margin:nNN . 2028, 2037, 2081, 2081
__xeCJK_check_family:V 2431, 2543
__xeCJK_check_family:n 2479, 2479, 2488
__xeCJK_check_num_range:nnNN 500, 500, 521, 2985
__xeCJK_check_single_save:N 1276, 1283, 1306, 1317, 1320, 1329, 1353, 1354, 1365, 1366, 1368, 1374, 1377, 1387, 1390
__xeCJK_check_single_space:NN . . . 1318, 1328, 1360, 1360
__xeCJK_class_csname:n 332, 335, 338, 343, 346, 351, 351, 352, 472, 475, 1482, 1500, 3605
__xeCJK_clear_Boundary_and_CJK_toks:n 590, 591
__xeCJK_copy_family:nn 2572, 2572, 2591
__xeCJK_copy_family:xx 2634
__xeCJK_copy_sub_family:n 2534, 2541
__xeCJK_dim_max:nn 3526, 3534, 3538

__xeCJK_dim_min:nn 3527, 3535, 3539
__xeCJK_document_left_hook: 88, 94
__xeCJK_document_right_hook: 90, 96
__xeCJK_error:n 60, 62, 1552
__xeCJK_error:nx 60, 63, 333, 344, 1488, 1558, 2142, 2171, 3326, 3518
__xeCJK_fallback_save_CJKsymbol:N 2205, 2212, 2220, 2235, 2244
__xeCJK_family_csname:n 2467, 2558, 2637, 2645, 2645, 2663
__xeCJK_family_default_wrap:n ... 2893, 2907, 3385, 3388
__xeCJK_family_nfss_csname:n 2484, 2587, 2588, 2645, 2646, 2647, 2651, 2661
__xeCJK_family_unknown_warning:n 2706, 2706, 2717
__xeCJK_family_unknown_warning:x 2682, 2694, 2704
__xeCJK_family_use:x 2600, 2609, 2641, 2645, 2647, 2680, 2703
__xeCJK_font_csname:n . 2593, 2594, 2606, 2608, 2628, 2631
__xeCJK_fontspec:nnn 2790, 2797
__xeCJK_fontspec:xnn 2785
__xeCJK_get_sub_features:Vn 2270
__xeCJK_get_sub_features:nn 2324, 2330, 2330, 2360
__xeCJK_get_sub_features:w 2330, 2334, 2345
__xeCJK_gobble_CJKfamily: 2697, 2697, 3665
__xeCJK_gobble_CJKfamily:wn 2698, 2699
__xeCJK_group_begin: 3628, 3628, 3636, 3667
__xeCJK_group_end: 3628, 3629, 3637, 3668
__xeCJK_gset_family_cs:x 2432, 2465, 2465
__xeCJK_gset_family_nfss_cs:xx .. 2474, 2565, 2645, 2648
__xeCJK_info:nxx 60, 68, 2619
__xeCJK_int_until_do:wn 230, 231, 232, 233
__xeCJK_kerning_width_or_ratio:nNN 2033, 2034, 2052, 2052
__xeCJK_listings_CJK_toks: 4218
__xeCJK_listings_CJK_toks_hook: 4240, 4248, 4248
__xeCJK_listings_append:nN 4288, 4288, 4302, 4314, 4329, 4350, 4370
__xeCJK_listings_breaklines_toks: ... 4218, 4248, 4262
__xeCJK_listings_escape:N 4453, 4454, 4455
__xeCJK_listings_initial_hook: 4202, 4211, 4211
__xeCJK_listings_inline_group:n 4438, 4442
__xeCJK_listings_inline_group:w 4429, 4436, 4441
__xeCJK_listings_inside_convert:nw .. 4429, 4429, 4435
__xeCJK_listings_output_IVS: 4208, 4402, 4402
__xeCJK_listings_peek_active_loop:TF 4414, 4414, 4423, 4428
__xeCJK_listings_process_CJK:nN 4251, 4253, 4255, 4259, 4276, 4282
__xeCJK_listings_process_Default:N .. 4237, 4276, 4276
__xeCJK_listings_process_FullLeft:nN . 4267, 4316, 4331
__xeCJK_listings_process_FullRight:nN 4269, 4316, 4352
__xeCJK_listings_process_IVS:nN 4239, 4395, 4395
__xeCJK_listings_process_breaklines_CJK:nN 4265, 4273, 4316, 4316
__xeCJK_listings_process_letter:nN 4285, 4293, 4293, 4400
__xeCJK_listings_process_other:nN ... 4286, 4293, 4304
__xeCJK_listings_toks_hook: 4203, 4221, 4221
__xeCJK_make_under_symbol:n 4086, 4114, 4139, 4139
__xeCJK_margin_width_or_ratio:n 1865, 1866, 1936, 1957, 1957
__xeCJK_math_robust:N 3447, 3447, 3491, 3492, 3493, 3494, 3495, 3502, 3503
__xeCJK_math_robust:NN 3456, 3457, 3459
__xeCJK_math_robust_aux:NN 3448, 3449
__xeCJK_msg_def_family_map:n 2722, 2724, 2938, 2945
__xeCJK_msg_family_map:n 2254, 2490, 2721, 2735, 2943, 2944, 3226
__xeCJK_msg_new:nn 60, 60, 353, 1492, 1546, 2147, 2161, 2252, 2489, 2625, 2719, 2852, 2934, 2941, 3222, 3330, 3375, 3508
__xeCJK_msg_new:nnn 61, 3485
__xeCJK_nobreak_ccglue: 3044, 3052, 3153
__xeCJK_nobreak_ecglue: 3048, 3054, 3154
__xeCJK_nobreak_hskip:n 3035, 3036, 3049, 3050, 3056
__xeCJK_nobreak_skip: 3009, 3021, 3038, 3100
__xeCJK_nobreak_skip_zero: . 3005, 3013, 3021, 3021, 3099
__xeCJK_original_kerning_margin:NN 1981, 2003, 2003, 2064
__xeCJK_parse_font_shape: 2430, 2491, 2491
__xeCJK_peek_after_do:w 268, 279
__xeCJK_peek_catcode_false:w 240, 259, 264
__xeCJK_peek_catcode_ignore_spaces_branches:w 243, 245, 250
__xeCJK_peek_catcode_true:w 239, 256, 263
__xeCJK_peek_ignore_spaces_branches:w .. 270, 272, 276
__xeCJK_prop_put_aux:n 2446, 2461
__xeCJK_prop_put_aux:nn 2446, 2463
__xeCJK_punct_bound_rule:NN 924, 924, 959, 969, 1125, 3970, 3995
__xeCJK_punct_breakable_kern:NN 1164, 1164, 1589
__xeCJK_punct_breakable_kern:n 1167, 1171, 3036, 3050, 3064, 3076, 3811
__xeCJK_punct_dim_csname:nn 1662, 1667, 1671
__xeCJK_punct_dim_csname:nnn 1664, 1669, 1673, 1750, 1782
__xeCJK_punct_glue:NN 942, 942, 961, 971, 981, 989, 1005, 1019, 1085, 1086, 1124, 1215, 3923, 3937, 3950, 3969, 3997, 4023, 4037
__xeCJK_punct_hskip:n 944, 951, 3035, 3049, 3062, 3074, 3810
__xeCJK_punct_if_long:NT 2111
__xeCJK_punct_if_long:NTF 1118, 3963
__xeCJK_punct_if_long_p:N 4360
__xeCJK_punct_if_middle:NTF 957, 967, 1121, 1812, 1828, 1852, 1877, 1890, 1921, 3966, 3992
__xeCJK_punct_if_mixed_width:NTF 1815, 1864
__xeCJK_punct_if_right:N 1648
__xeCJK_punct_if_right:NF 1990
__xeCJK_punct_if_right:NT 1992
__xeCJK_punct_if_right:NTF 2008, 2011, 2087, 2090
__xeCJK_punct_kern:NN ... 952, 952, 1136, 1145, 1160, 1176
__xeCJK_punct_kern:n 939, 940, 3809
__xeCJK_punct_min_bound:NN 2021, 2024, 2067, 2067
__xeCJK_punct_nobreak_kern:NN ... 1158, 1158, 1163, 1594
__xeCJK_punct_offset:NN 938, 938, 980, 988, 1006, 1020, 1026, 1209, 3924, 3951, 4022, 4036
__xeCJK_punct_rule:NN 931, 931, 979, 986, 1000, 1013, 1031, 1166, 1169, 1208, 4019, 4033
__xeCJK_punct_width_or_ratio:nN 1813, 1816, 1817, 1911, 1944, 1944
__xeCJK_remove_duplicate_keys:N 2427, 2443, 2443
__xeCJK_reset_char_class:n 3126, 3127, 3128, 3129, 3130, 3161
__xeCJK_reset_shipout_skip: 3023, 3040, 3058, 3058
__xeCJK_restore_shipout_CJKsymbol: 4089, 4117, 4123, 4123
__xeCJK_save_CJK_class:n 469, 474, 1537
__xeCJK_save_FullRight_check: 1188, 1198
__xeCJK_save_FullRight_symbol:N 1189, 1199, 1220
__xeCJK_save_family_info: 2433, 2521, 2521
__xeCJK_save_punct_dim:nnn 1670, 2107, 2109
__xeCJK_save_punct_dim:nnnn 1672, 1754, 1755, 1756, 1757, 1758, 1759, 1760, 1786, 1787, 1872, 1873, 1886, 1940, 1941, 1984, 1994, 2098, 2100, 2113
__xeCJK_save_under_CJKsymbol:N 4087, 4093, 4115, 4128, 4156
__xeCJK_set_char_class_aux:Nnw 481, 490, 496, 2978

__xeCJK_set_char_class_eq:nn 529, 529, 3115, 3116, 3117, 3118, 3119
__xeCJK_set_family_initial: 2399, 2399, 2421
__xeCJK_set_listings_escape: 4432, 4445, 4448, 4452
__xeCJK_set_others_toks:n 3591, 3599, 3603
__xeCJK_set_special_punct:nn 1597, 1600, 1603, 1618
__xeCJK_set_sub_block_family: 2434, 2528, 2528
__xeCJK_set_sub_class_toks:nn 1503, 1509, 1509
__xeCJK_set_verb_exspace: 3094, 3169, 3169
__xeCJK_set_verb_exspace:n 3186, 3192, 3192
__xeCJK_set_verb_scale:nn 3201, 3211, 3211
__xeCJK_set_visible_space_size:n 3259, 3263
__xeCJK_shipout_CJKecglue: 3061, 3073, 3121, 3137
__xeCJK_shipout_CJKglue: 3060, 3072, 3120, 3136
__xeCJK_shipout_boundary:w 3025, 3029, 3123, 3139
__xeCJK_shipout_check_for_glue: . 3024, 3028, 3122, 3138
__xeCJK_shipout_punct_breakable_kern:n ... 3064, 3076
__xeCJK_shipout_punct_hskip:n 3062, 3074
__xeCJK_space_skip_scale:nnn 164, 168, 181, 185, 195
__xeCJK_special_punct_seq:n 1614, 1617, 1620, 1622, 1626, 1633, 1636, 1645
__xeCJK_special_punct_tl:nN 1615, 1621, 1625, 1635, 1644, 1658
__xeCJK_sub_restore_or_cancel:n 1478, 1478, 1491
__xeCJK_sub_restore_or_cancel:x 1465, 1474
__xeCJK_sub_special_punct:nn 1599, 1602, 1605, 1640
__xeCJK_swap_cs_aux:w 133, 135, 136
__xeCJK_switch_font:nn 1518, 1523, 1532, 1534, 1541, 2615, 2615, 3908, 3913
__xeCJK_tl_remove_outer_braces:w 119, 121
__xeCJK_tmp:w 4449, 4460
__xeCJK_token_value_charcode:w 285, 288, 300
__xeCJK_ulem_Boundary_and_Default: 3806, 3851, 3851, 3859
__xeCJK_ulem_Boundary_and_FullLeft_glue:N 3817, 3929, 3929, 3940
__xeCJK_ulem_Boundary_and_NormalSp: 3807, 3861, 3861, 3865
__xeCJK_ulem_CJK_and_Boundary_aux: 3812, 3867, 3867, 3875, 4106
__xeCJK_ulem_CJK_and_CJK:N 3805, 3883, 3883, 3892
__xeCJK_ulem_CJK_and_FullLeft_glue:N 3815, 3942, 3942, 3955
__xeCJK_ulem_CJK_and_FullRight_glue:N 3816, 3978, 3978, 3986
__xeCJK_ulem_Default_and_FullLeft_glue:N 3813, 3917, 3917, 3927
__xeCJK_ulem_Default_and_FullRight_glue:N 3814, 3957, 3957, 3976
__xeCJK_ulem_FullLeft_and_CJK: .. 3802, 4006, 4006, 4013
__xeCJK_ulem_FullLeft_and_Default: 3801, 3988, 3988, 4004
__xeCJK_ulem_FullRight_and_CJK: . 3804, 4029, 4029, 4043
__xeCJK_ulem_FullRight_and_Default: 3803, 4015, 4015, 4027
__xeCJK_ulem_between_CJK_blocks:nnN 3826, 3828, 3832, 3900, 3900
__xeCJK_ulem_ccglue: 3888, 3905, 3949, 3964, 4038, 4067, 4073
__xeCJK_ulem_class_group_begin: 3889, 3894, 3894, 3953, 3984, 4011, 4041, 4063
__xeCJK_ulem_fix_penalty: 3808, 3877, 3877, 3881
__xeCJK_ulem_glue:n 3781, 3783, 4067, 4067
__xeCJK_ulem_hskip:n 3872, 4048, 4054, 4062, 4067, 4070, 4075
__xeCJK_ulem_initial: 3765, 3798, 3798
__xeCJK_ulem_leader_type: 3760, 3796, 3797
__xeCJK_ulem_punct_breakable_kern:n 3811, 4057, 4057, 4065
__xeCJK_ulem_punct_hskip:n 3810, 4045, 4045, 4049
__xeCJK_ulem_punct_kern:n 3809, 4051, 4051, 4055
__xeCJK_ulem_restore_CJK_and_Boundary: 4084, 4097, 4101
__xeCJK_ulem_skip_punct_begin: 3762, 3793, 3793, 3922, 3934, 3948, 3962
__xeCJK_ulem_skip_punct_end: 3763, 3793, 3795, 4000, 4024, 4039
__xeCJK_ulem_swap_cs:NN 3800, 3837, 3841
__xeCJK_under_CJKsymbol:N 4088, 4116, 4153, 4153
__xeCJK_update_cs_case_tl:NNnn .. 1230, 1236, 1241, 1245
__xeCJK_update_family:nn 2470, 3545, 3553
__xeCJK_update_inline_env_case_tl: 1423, 1432, 1438, 1442, 1442
__xeCJK_update_url_font: 3424, 3424, 3433
__xeCJK_use_punct_dim:nn 1666, 1831, 1843, 1879, 1880, 1925, 1929, 1953, 2041, 2042, 2092, 2093, 2110
__xeCJK_use_punct_dim:nnn 927, 934, 939, 946, 947, 948, 953, 1168, 1668, 1765, 1768, 1924, 1987, 1988, 1997, 1998, 2007, 2010, 2072, 2073, 2077, 2078, 2086, 2089, 2104, 2105
__xeCJK_verb_CJK_and_Boundary:w 3157, 3159
__xeCJK_verb_addon: 3095, 3110
__xeCJK_verb_font_hook: 3001, 3019
__xeCJK_warning:n 64, 2847
__xeCJK_warning:nx 60, 65, 2156, 2713, 2826, 2909, 3371, 3372, 3373
__xeCJK_warning:nxx 66, 2485, 2927, 3214, 3475, 3481
__xeCJK_warning:nxxx 67, 2241
__xeCJK_zero_glue: 973, 975, 1049, 1057, 3936
__xunadd_add_accent:nnNN 4763, 4771, 4776, 4776
__xunadd_add_circle:nn 4789, 4802, 4802
__xunadd_begin_csname:n 4863, 4868, 4869
__xunadd_begin_hook:nn 4670, 4757, 4865, 4865
__xunadd_chardef:Nn 4699, 4701
__xunadd_chardef:cn 4694
__xunadd_check_slot:n 4648, 4676, 4676, 4696
__xunadd_declare_accent:NNNnnn 4748, 4754
__xunadd_declare_accent:NNNxxn 4744
__xunadd_declare_accent:Nnnn 4725, 4732, 4736, 4736, 4753, 4885
__xunadd_declare_accent:cnnn 4726, 4733
__xunadd_declare_character:NNnn 4666, 4666, 4675
__xunadd_declare_character:NNxn 4652
__xunadd_declare_character:Nnn 4543, 4544, 4545, 4546, 4547, 4548, 4628, 4644, 4644, 4656, 4884
__xunadd_declare_character:cnn 4629
__xunadd_declare_composite:Nnnn 4687, 4691, 4691, 4702, 4886, 4887, 4888, 4889, 4890, 4891, 4892, 4893, 4894, 4895, 4896, 4897, 4898, 4899, 4900, 4901, 4902, 4903, 4904, 4905, 4906, 4907, 4908, 4909, 4910, 4911, 4912, 4913, 4914, 4915, 4916, 4917, 4918, 4919, 4920, 4921, 4922, 4923, 4924, 4925, 4926, 4927, 4928, 4929, 4930, 4931, 4932, 4933, 4934, 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943, 4944, 4945, 4946, 4947, 4948, 4949, 4950, 4951, 4952, 4953, 4954, 4955, 4956, 4957, 4958, 4959, 4960, 4961, 4962, 4963, 4964, 4965, 4966, 4967, 4968, 4969, 4970, 4971, 4972, 4973, 4974, 4975, 4976, 4977, 4978, 4979, 4980, 4981, 4982, 4983, 4984, 4985, 4986, 4987, 4988
__xunadd_declare_composite:cnnn 4688
__xunadd_declare_composite_command:Nnnn 4711, 4715, 4715, 4721
__xunadd_declare_composite_command:cnnn 4712
__xunadd_declare_math_as_UTF_text:n 4555, 4569
__xunadd_declare_math_symbols: 4541
__xunadd_declare_math_symbols:n 4526, 4541
__xunadd_declare_textcircled:n 4525, 4879, 4882
__xunadd_end_csname:n 4864, 4874, 4875
__xunadd_end_hook:nn 4673, 4764, 4865, 4872

__xunadd_glyph_if_exist:n 4591
__xunadd_glyph_if_exist:nTF 4591, 4671, 4770, 4780, 4788, 4792, 4795
__xunadd_glyph_if_exist_p:n 4591
__xunadd_if_csname:n 4612
__xunadd_if_csname:nTF 4598, 4605, 4612, 4627, 4686, 4710, 4724, 4731
__xunadd_provide_text_command_default:N 4646, 4657, 4657
__xunadd_reload:N 4509, 4514
__xunadd_restore_hbar: 4625, 4633, 4633
__xunadd_restore_hbar:N 4638, 4643
__xunadd_restore_hbar:c 4636
__xunadd_set_cmd_hook:nnn 4837, 4847, 4851, 4851
__xunadd_set_cmd_hook_aux:Nnwn 4857, 4861
__xunadd_text_character:nN 4667, 4668
__xunadd_text_composite:nnNNn 4751, 4755, 4755
__xunadd_text_composite_aux:NnNNn 4766, 4775
__xunadd_text_composite_aux:cnNNn 4760
__xunadd_tmp:w 4494, 4497, 4516, 4530
__xunadd_undeclare_composite:Nnnn . . . 4606, 4607, 4610

\␣ . . 1495, 1548, 3332, 3378, 3379, 3506, 3768, 3775, 3845, 4462

A

\add@accent 4796
\addCJKfontfeature 2808, 2854
\addCJKfontfeatures 6, 2799, 2803, 2808
\AddEverypageHook 99
\addto@hook 3752
\AfterEndPreamble 84
\AfterPreamble 83
KaiMingPunct 4
KaiMingPunct+ 4
KaiMingPunct- 4
FallBack 7
\AllowBreakBetweenPuncts 1583
Mapping 5
\arraystretch 4170
\AssignTemplateKeys 1747
\AtBeginDocument 87, 4586
\AtBeginUTFCommand 3507, 3636, 4829, 4829, 4879
\AtEndOfPackage 600, 1553, 3248, 3692, 4500, 4703
\AtEndPreamble 82
\AtEndUTFCommand 3637, 4829, 4839
\AutoFakeBold 2282, 2368
\AutoFakeSlant 2282, 2368
\AutoFallBack 2198

B

\begin 3356
\bfdefault 2966, 2969
\BODY 4178, 4184, 4191
\BoldFont 2363
\bool_gset_false:N 3286, 4214, 4407
\bool_gset_true:N 2288, 2296, 3289, 3294, 4398
\bool_if:NF 102, 1267, 1462, 1485, 3088, 3112, 3755
\bool_if:nF 3496, 4336, 4659
\bool_if:NT 882, 1471, 1965, 2495, 2507, 2931, 3235, 3368, 3507, 3736, 3737, 4083, 4090, 4092, 4096, 4405, 4507, 4511, 4552
\bool_if:nT 734, 886, 2828, 2836, 3432, 3674, 4357
\bool_if:NTF 758, 841, 846, 878, 1258, 1314, 1316, 1327, 1335, 1347, 1352, 1800, 1835, 1848, 1900, 1915, 1933, 1979, 2020, 2023, 3068, 3132, 3759, 4296, 4307, 4319, 4334, 4355, 4375, 4386, 4785
\bool_if:nTF 123, 502, 632, 640, 648, 679, 687, 720, 727, 743, 751, 848, 1041, 1052, 1062, 1298, 1341, 1972, 3645
\bool_if_p:n 4617
\bool_new:N 57, 111, 265, 598, 829, 1459, 1611, 2278, 2279, 2414, 2415, 3106, 3167, 3300, 3786, 3792, 4413, 4562, 4881
\bool_set_eq:NN 2406, 2407
\bool_set_false:N 241, 816, 1260, 1473, 1593, 2377, 2391, 3791, 4310, 4378, 4389, 4556, 4787
\bool_set_true:N 104, 248, 596, 811, 821, 1255, 1464, 1588, 2373, 2380, 2387, 2394, 3114, 3757, 3789, 4210, 4300, 4326, 4347, 4367, 4566, 4880
\Boundary 358
\box_new:N 55, 4100
\box_set_to_last:N 1040
\box_use:N 4148, 4155
\box_use_clear:N 1046, 1047
\box_wd:N 218, 1044, 4147, 4181

C

\c__xeCJK_CJ_chars_clist 402, 402, 418
\c__xeCJK_CJK_chars_clist 421, 421, 549, 2973
\c__xeCJK_CL_chars_clist 386, 386, 414
\c__xeCJK_config_ext_tl 3739, 3741
\c__xeCJK_encoding_tl . . 2653, 2963, 2966, 2968, 3367, 3367
\c__xeCJK_EX_chars_clist 399, 399, 416
\c__xeCJK_FullLeft_chars_clist 381, 381, 544, 2973
\c__xeCJK_FullRight_chars_clist 412, 412, 545, 2975
\c__xeCJK_HalfLeft_chars_clist 367, 367, 542
\c__xeCJK_HalfRight_chars_clist 367, 369, 543
\c__xeCJK_IS_chars_clist 401, 401, 417
\c__xeCJK_IVS_chars_clist 457, 457, 551
\c__xeCJK_left_tl 961, 971, 993, 1000, 1005, 1006, 1010, 1013, 1019, 1020, 1024, 1026, 1031, 1085, 1086, 1125, 1134, 1152, 1169, 1581, 1581, 1759, 1769, 1988, 1998, 2012, 2072, 2077, 2088, 2098, 2104, 3923, 3924, 3937, 3950, 3951, 3970, 3994, 3997
\c__xeCJK_math_family_tl 2962, 2964, 2966, 2969
\c__xeCJK_math_tl . 2949, 2950, 2953, 2954, 2963, 2968, 2971
\c__xeCJK_mono_letter_int 3092, 3105
\c__xeCJK_NormalSpace_chars_clist 367, 371, 550
\c__xeCJK_NS_chars_clist 394, 394, 415
\c__xeCJK_OP_chars_clist 372, 372, 383
\c__xeCJK_package_ext_tl 13, 16, 36
\c__xeCJK_PO_chars_clist 411, 411, 419
\c__xeCJK_PR_chars_clist 379, 379, 384
\c__xeCJK_punct_style_plain_tl 1752, 1784, 2137, 2146, 4206
\c__xeCJK_right_tl 959, 969, 979, 980, 981, 986, 988, 989, 1090, 1101, 1111, 1124, 1143, 1166, 1174, 1207, 1208, 1209, 1215, 1581, 1582, 1760, 1769, 1987, 1997, 2009, 2073, 2078, 2091, 2100, 2105, 3969, 3995, 4019, 4022, 4023, 4033, 4036, 4037
\c_catcode_letter_token 1290, 1312
\c_catcode_other_space_tl 3242
\c_eleven 209, 635, 682, 737, 1050, 1256
\c_four 156, 177, 3596
\c_fourteen 3007, 3011
\c_group_begin_token 595
\c_group_end_token 599
\c_math_toggle_token 839, 876, 1372
\c_max_dim 1678, 1684, 1690, 1692, 1700, 1703, 1705, 1707, 1826, 1854, 1919, 1946, 1950, 1959, 2027, 2054, 2059
\c_minus_one 1043, 1049
\c_nan_fp 1691, 1704, 1706
\c_one 325, 359, 1038, 1078, 2099, 2786, 3596, 3935, 4290, 4328
\c_one_fp 1679, 1685, 1708
\c_one_thousand 150, 204
\c_seven 171, 186
\c_space_tl 817, 3857, 3864
\c_space_token 247, 274
\c_ten 730, 746
\c_ten_thousand 70

\c_thirteen 1064, 1073
\c_three 155, 176, 361, 2101, 3594, 4077, 4339, 4369
\c_true_bool 3217
\c_two 154, 164, 170, 360, 1832, 1859, 1880, 3091, 3195, 3202, 3260, 3268, 3275, 4251, 4253, 4255, 4259, 4265, 4267, 4269, 4273, 4321, 4338, 4349, 4359
\c_two_hundred_fifty_five 362
\c_two_hundred_fifty_six 3718
\c_xeCJK_math_fam_int 2971, 2979, 3427
\c_xeCJK_space_skip_tl 148, 148, 636, 683, 738
\c_zero 69, 326, 358, 1065, 1075, 1079, 1080, 1674, 1675, 1711, 2979, 3086, 3239, 3559, 4239
\c_zero_dim 928, 929, 935, 936, 1697, 1709, 1754, 1755, 1756, 1757, 1758, 1759, 1760, 1786, 1787, 1875, 1888, 2117, 3196, 3339, 4142, 4172, 4173, 4818, 4820
\c_zero_fp 1693, 1949, 2030, 2057
\c_zero_skip .. 152, 161, 166, 183, 976, 3042, 3046, 3145, 3872
\char_set_catcode:nn 3687, 4510
\char_set_catcode_active:N 4459
\char_set_catcode_ignore:n 327
\char_set_catcode_letter:n 516, 3685, 4508
\char_set_lccode:nn 3729, 4648, 4739, 4740
\char_value_catcode:n 3687, 4510
\CheckFullRight 1181
\CheckSingle 1269
\CJK 358
\CJK@@UL 3787, 3787, 3790
\CJK@hundredmillion 3723
\CJK@ifundefined 3678, 3679
\CJK@nest 4136
\CJK@postUnderdot 4095, 4136
\CJK@preUnderdot 4085
\CJK@tenthousand 3722
\CJK@UL 3787, 3787, 3788
\CJK@underdotBox 4113, 4148
\CJK@underdotSkip 4086
\CJK@UnicodeEnc 3714
\CJKaddEncHook 3712, 3712
\CJKecglue 628, 645, 653, 673, 692, 732, 749, 801, 805, 812, 822, 824, 843, 879, 3032, 3045, 3047, 3048, 3061, 3073, 3121, 3137, 3148, 3154, 3156, 3160, 3778, 3782
xCJKecglue 3
\CJKf@global 4135
\CJKfamily 6, 2666, 2666, 2698
\CJKfamilydefault 6, 2239, 2247, 2637, 2638, 2867, 2870, 2894, 2896, 2904, 2909, 2911, 2913, 2914, 2918, 2924, 2928, 2930, 2956, 2957, 3381, 3390, 3391, 3395, 3550
\CJKfilltwosides 4158
\CJKfixedspacing 3168
\CJKfontspec 6, 2766, 2774
\CJKglue 722, 725, 759, 792, 796, 896, 982, 1018, 1119, 3031, 3041, 3043, 3044, 3060, 3072, 3120, 3136, 3147, 3153, 3771, 3780, 4161, 4169, 4409
\CJKmath 2947
\CJKnospace 3409, 3410
\CJKpunctsymbol 1001, 1014, 1032, 1138, 1156, 1204, 3335, 3336
\CJKrmdefault 6, 2728, 2739, 2747, 2900, 2914, 2917, 2918, 3381, 3381, 3392, 3547
\CJKsetecglue 3407, 3407, 3408
\CJKsfdefault 6, 2729, 2740, 2753, 2901, 3381, 3382, 3548
\CJKspace 830, 3409, 3409
\CJKsymbol 621, 716, 896, 898, 900, 1522, 1523, 2203, 2205, 2206, 2211, 2212, 3335, 3335, 3890, 3909, 3914, 4087, 4088, 4093, 4115, 4116, 4128
\CJKttdefault 6, 2730, 2741, 2758, 2902, 3381, 3383, 3549
\CJKunderanysymbol 4110, 4110
\CJKunderdot 4081, 4081
\clist_clear:N 2333, 2405, 2447, 2818
\clist_concat:NNN 2425, 2839, 2842, 2878, 2972, 2974
\clist_const:Nn 367, 369, 371, 372, 379, 386, 394, 399, 401, 402, 411, 421, 457
\clist_const:Nx 381, 412
\clist_gconcat:ccN 484
\clist_gconcat:NNN 498
\clist_get:NNF 4484
\clist_gput_left:NV 4478
\clist_gput_left:Nx 4481
\clist_gput_right:Nx 4523
\clist_gremove_duplicates:N 4531
\clist_gset:Nn 1613, 2801
\clist_map_function:nN 474
\clist_map_inline:cn 532, 3164
\clist_map_inline:Nn 486, 1616, 1653, 2976, 4518
\clist_map_inline:nn 27, 32, 613, 625, 660, 903, 919, 921, 1427, 1436, 1480, 1538, 2267, 2819
\clist_new:c 336, 347
\clist_new:N 59, 1612, 2359, 2413, 2439, 2799, 2849, 2850, 4476, 4533
\clist_put_left:Nn 2271
\clist_put_right:No 2452
\clist_put_right:Nx ... 2454, 2497, 2502, 2509, 2514, 2880
\clist_remove_all:Nn 2319, 2553, 2816
\clist_remove_duplicates:N 4504
\clist_set:Nn 483, 2354, 2423, 2814
\clist_set:Nx 4503
\clist_set_eq:NN 4489
\coffin_attach:NnnNnnnn 4815
\coffin_dp:N 4811
\coffin_ht:N 4811
\coffin_new:N 4823, 4824
\coffin_resize:Nnn 4808
\coffin_typeset:Nnnnn 4819
\coffin_wd:N 4807, 4813
\cs:w 141, 3676
\cs_end: 141, 306, 1658, 3676
\cs_generate_variant:Nn 311, 498, 499, 557, 567, 573, 1218, 1393, 1403, 1491, 1508, 1577, 1795, 2122, 2360, 2361, 2440, 2460, 2488, 2591, 2658, 2717, 2796, 2797, 2851, 2889, 2890, 4643, 4656, 4675, 4701, 4702, 4721, 4753, 4754, 4775
\cs_gset_eq:cc 2586
\cs_gset_eq:NN 130, 140, 3249, 3534, 3535, 4087, 4640
\cs_gset_nopar:Npx 92
\cs_gset_protected_nopar:cpx 2467, 2557, 2651, 4574
\cs_gset_protected_nopar:Npx 3437, 3465, 3471
\cs_if_eq:NNF 1186, 1274, 2203
\cs_if_eq:NNT 1196, 1282, 2211
\cs_if_eq:NNTF 3469, 3691, 4455
\cs_if_exist:cTF 2636, 2661, 4520, 4571, 4758
\cs_if_exist:NF 1544, 3678
\cs_if_exist_p:c 4661
\cs_if_exist_p:N 3432, 3675
\cs_if_exist_use:cF 2597, 2606, 2628, 3252, 4672, 4868, 4869, 4874, 4875
\cs_if_exist_use:cTF 1802, 1902, 2663, 4782, 4799
\cs_if_free:cF 2965, 4635
\cs_if_free:NF 3435, 4588
\cs_if_free:NTF 3679
\cs_if_free_p:c 4662
\cs_new:Npn 116, 121, 4861
\cs_new_eq:cc 4573
\cs_new_eq:cN 352
\cs_new_eq:NN 297, 302, 599, 951, 1130, 1163, 1171, 2614, 2750, 2808, 3168, 3385, 3408, 3526, 3527, 3553, 3554, 3628, 3629, 3642
\cs_new_nopar:cpx 4583
\cs_new_nopar:Npn 112, 195, 282, 284, 288, 300, 309, 351, 475, 558,

1204, 1614, 1615, 1662, 1664, 1666, 1668, 1944, 1957, 2003, 2015, 2052, 2067, 2081, 2123, 2594, 2645, 2646, 2647, 2724, 2735, 3335, 3336, 3843, 4243, 4676, 4863, 4864
\cs_new_protected:Npn 74, 76, 78, 108, 127, 129, 131, 230, 232, 236, 266, 868, 1221, 1245, 1442, 3712, 4123, 4139, 4414, 4429, 4442, 4452, 4454, 4514, 4610, 4644, 4657, 4666, 4668, 4691, 4699, 4715, 4736, 4748, 4755, 4766, 4776, 4802, 4851, 4865, 4872
\cs_new_protected_nopar:Npn 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 88, 90, 114, 138, 205, 245, 272, 330, 341, 469, 481, 500, 519, 529, 555, 560, 562, 568, 574, 580, 589, 591, 593, 630, 677, 718, 773, 779, 784, 837, 866, 874, 896, 917, 924, 931, 938, 940, 942, 952, 955, 965, 975, 977, 984, 991, 1003, 1008, 1016, 1022, 1034, 1088, 1099, 1109, 1116, 1131, 1140, 1150, 1158, 1164, 1172, 1205, 1219, 1288, 1310, 1360, 1370, 1379, 1394, 1478, 1498, 1509, 1618, 1629, 1640, 1670, 1672, 1748, 1780, 1796, 1896, 1968, 2096, 2217, 2228, 2260, 2308, 2330, 2345, 2399, 2418, 2441, 2443, 2461, 2463, 2465, 2479, 2491, 2521, 2528, 2541, 2572, 2595, 2604, 2615, 2626, 2648, 2687, 2697, 2700, 2706, 2779, 2790, 2809, 2857, 2951, 2983, 3001, 3021, 3038, 3052, 3054, 3056, 3058, 3110, 3159, 3161, 3169, 3192, 3211, 3233, 3250, 3256, 3263, 3424, 3447, 3449, 3459, 3557, 3603, 3726, 3753, 3793, 3795, 3798, 3837, 3851, 3861, 3867, 3877, 3883, 3894, 3900, 3917, 3929, 3942, 3957, 3978, 3988, 4006, 4015, 4029, 4045, 4051, 4057, 4067, 4073, 4075, 4101, 4153, 4211, 4221, 4248, 4262, 4276, 4282, 4288, 4293, 4304, 4316, 4331, 4352, 4395, 4402, 4436, 4541, 4569, 4633, 4638, 4882
\cs_new_protected_nopar:Npx 2771
\cs_set:Npn 4449
\cs_set_eq:NN 128, 133, 134, 135, 238, 812, 817, 824, 872, 1188, 1189, 1190, 1191, 1198, 1199, 1276, 1277, 1283, 1589, 1594, 2205, 2206, 2212, 2470, 2698, 2893, 3024, 3025, 3028, 3029, 3034, 3035, 3036, 3044, 3048, 3049, 3050, 3060, 3061, 3062, 3063, 3072, 3073, 3074, 3075, 3120, 3121, 3122, 3123, 3136, 3137, 3138, 3139, 3153, 3154, 3156, 3157, 3679, 3719, 3760, 3768, 3769, 3770, 3775, 3776, 3777, 3787, 3796, 4088, 4093, 4115, 4116, 4128, 4135, 4161, 4169, 4218, 4313, 4409, 4428, 4435, 4441, 4494, 4495, 4497, 4516, 4517, 4530, 4581
\cs_set_nopar:Npn 3643, 3717
\cs_set_protected:cpn 4717, 4853
\cs_set_protected_nopar:Npn 796, 805, 822, 3780, 3782, 4373, 4384
\cs_set_protected_nopar:Npx 3018, 3124
\cs_to_str:N 2770, 3448
\cs_undefine:c 1621, 1644, 2484, 4611
\cs_undefine:N 136, 2907, 3538, 3539, 4641
\curr@fontshape 3171, 3174, 3177, 3207, 3252, 3258
\CurrentOption 3746, 4198, 4478

D

\DeclareEncodedCompositeAccents 4729, 4729
\DeclareEncodedCompositeCharacter 4722, 4722
\DeclareExpandableDocumentCommand 2699
\DeclareInstance 2158
\DeclareObjectType 1674
\DeclareOption 3746, 4198, 4477
\DeclareSymbolFont 2963
\DeclareSymbolFontAlphabet 3443
\DeclareTemplateCode 1711
\DeclareTemplateInterface 1675
\DeclareTextCommand 4667, 4750
\DeclareUTFcharacter 4622, 4622
\DeclareUTFcomposite 4684, 4684
\DeclareUTFCompositeCommand 4705, 4706, 4708, 4708
\DeclareUTFmathsymbols 4550, 4550
\def 3719
\Default 358
\defaultCJKfontfeatures 6, 2799, 2800, 2802, 3363
\dim_compare:nNnF 3268
\dim_compare:nNnTF 1826, 1854, 1919, 1946, 1959, 2027, 2054, 2059, 3090, 3196, 4173, 4181
\dim_compare_p:nNn 1044
\dim_const:cn 776
\dim_eval:n . . . 197, 1671, 1673, 2005, 2083, 3272, 4160, 4183
\dim_gset:cn 776
\dim_if_exist:cTF 775
\dim_max:nn 1823, 1837, 1875, 1888, 1917, 2017, 2024, 2069, 3526, 3534
\dim_min:nn . . 1838, 1849, 1934, 1966, 2071, 2076, 3527, 3535
\dim_new:N 56, 1778, 1779, 4822
\dim_ratio:nn 3275
\dim_set:Nn 1764, 1766, 1798, 1821, 1898, 1913, 1970, 2102, 4806
\dim_set_eq:NN 4172
\dim_to_fp:n 3202, 3203
\dim_use:N 218, 2125, 3187, 3260
\document 92, 95

E

xeCJKactive 3
\EditInstance 2170
\else: 146, 225, 227, 252, 257, 278, 307, 316, 771, 1651, 1659, 3847, 4299, 4325, 4346, 4366, 4422, 4594, 4618
\EmboldenFactor 2282
\encodingdefault 4538
\end 3356
\endmath 3447, 3494
\endminipage 4164
\endtabular 4179, 4185, 4192
\ensuremath 3447, 3469, 3495
\EnvCS 1411
Verb 5
\etex_currentgrouplevel:D 3086
\etex_currentgrouptype:D 3007, 3011
\etex_dimexpr:D 1858, 1953, 1963, 2040, 2064
\etex_fontcharwd:D 2108, 3092, 3187
\etex_glueshrink:D 191
\etex_gluestretch:D 190
\etex_iffontchar:D 145, 4593
\etex_lastnodetype:D 209, 635, 682, 730, 737, 746, 1036, 1043, 1064, 1078, 1256, 3935
\etex_numexpr:D 4593, 4700
NewLineCS 3
NewLineCS+ 3
NewLineCS- 3
\exp_after:wN 140, 141, 211, 212, 216, 217, 233, 249, 250, 255, 256, 258, 259, 275, 276, 279, 285, 1302, 1303, 1305, 2334, 3185, 3186, 3187, 3259, 3260, 3426, 3427, 3581, 3582, 3583, 3584, 3676, 3846, 3848, 4438, 4439
\exp_args:Nc 335, 1655, 4600, 4607
\exp_args:NNc 3448
\exp_args:NNv 4664
\exp_args:Nnx 2158, 2170
\exp_args:No 1301, 3463
\exp_args:NV 115
\exp_args:Nx 3451
\exp_last_unbraced:Nf 118
\exp_not:c 906, 3067, 3131, 4577, 4578, 4581, 4583
\exp_not:N 254, 2474, 2562, 2563, 2567, 2653, 2654, 2655, 2900, 2901, 2902, 3388, 3392, 4160, 4509, 4576
\exp_not:n 565, 571, 2470, 2471, 2893, 3070, 3134
\exp_not:o 95, 2455, 3019, 3439, 3472
\exp_not:V 578, 586, 2341, 2342, 2455, 2472, 2473, 2503, 2515, 2884, 2885, 3391
\exp_stop_f: 145, 223, 226, 315, 941, 953, 2126

\ExplSyntaxOff 4507
\ExplSyntaxOn 4511
\extrarowheight 4171, 4172

F

\f@baselineskip 3278
\f@encoding 3647
\f@family 2654, 3228, 3229, 3241, 3266, 3544
\f@series 2594
\f@shape 2594
\f@size 2594, 3171, 3174, 3207, 3252, 3258, 3274
\FallBack 2362
\familydefault 2898, 3550
\fi: . . 146, 227, 228, 233, 260, 261, 280, 307, 316, 771, 1651, 1659, 3014, 3849, 4219, 4299, 4312, 4325, 4346, 4366, 4380, 4381, 4391, 4392, 4399, 4411, 4422, 4439, 4594, 4618
\file_input:n 3686, 4524
\fix@penalty 3554, 3556, 3880
\fontencoding 2653, 3265
\fontfamily 3542, 3542
\fontsize 3270
\fontspec_set_family:Nnn 2471
\fontspec_setup_maths: 3435, 3435, 3437, 3441
\fontspec_visible_space: 3249
\footnote 3357
\footnotemark 3357
\fp_compare:nNnTF 286, 1949, 2030, 2057
\fp_eval:n 198, 3215, 3216
\fp_gset:Nn 2289, 2297
\fp_new:N 2280, 2281, 2416, 2417, 3231, 4827
\fp_set:Nn 2381, 2395, 3213, 4826
\fp_set_eq:NN 2374, 2388, 2408, 2409
\fp_use:c 1952, 1962, 2063
\fp_use:N 1857, 2039, 2061, 2498, 2510, 3218, 4807
\FullLeft 358
\FullRight 358

G

\g__xeCJK_after_end_preamble_hook_tl 73, 79, 84, 91
\g__xeCJK_after_preamble_hook_tl 72, 77, 83, 87
\g__xeCJK_at_end_preamble_hook_tl 71, 75, 82, 89
\g__xeCJK_auto_fake_bold_bool 2278, 2285, 2288, 2406
\g__xeCJK_auto_fake_slant_bool . . . 2279, 2293, 2296, 2407
\g__xeCJK_base_class_seq 463, 463, 464, 1511
\g__xeCJK_CJK_class_seq 463, 468, 471, 590, 3606
\g__xeCJK_CJK_sub_class_seq 1451, 1451, 1527, 1536, 3819, 3821, 4231, 4256, 4270
\g__xeCJK_class_seq 328, 328, 337, 348, 464, 602
\g__xeCJK_config_bool 3286, 3289, 3294, 3300, 3737
\g__xeCJK_config_name_tl 3290, 3295, 3299, 3741
\g__xeCJK_default_features_clist 2426, 2799, 2801
\g__xeCJK_embolden_factor_fp . 2280, 2289, 2300, 2374, 2408
\g__xeCJK_family_font_name_prop 2255, 2264, 2481, 2518, 2519, 2523, 2544, 2547, 2579, 2708, 2811, 2859, 2866, 2908, 2921
\g__xeCJK_family_font_options_prop 2518, 2520, 2525, 2550, 2554, 2580, 2833, 2862, 2869, 2873
\g__xeCJK_family_int 2401, 2411, 2786
\g__xeCJK_family_name_prop 2483, 2518, 2518, 2562, 2578, 2650, 2960
\g__xeCJK_fontspec_prop 2781, 2792, 2798
\g__xeCJK_indent_bool 3304, 3368
\g__xeCJK_last_punct_tl 954, 954, 957, 959, 961, 967, 969, 971, 979, 980, 981, 986, 988, 989, 999, 1012, 1030, 1096, 1106, 1113, 1135, 1136, 1137, 1144, 1145, 1146, 1153, 1154, 1155, 1175, 1176, 1177, 1207, 1208, 1209, 1215, 3992, 3994, 3995, 3997, 4019, 4022, 4023, 4033, 4036, 4037
\g__xeCJK_listings_IVS_bool . 4214, 4398, 4405, 4407, 4413
\g__xeCJK_math_bool 2931, 2947, 3432
\g__xeCJK_new_class_seq 328, 329, 338, 3594, 3598
\g__xeCJK_non_CJK_class_seq 463, 465, 466, 901
\g__xeCJK_number_bool 3303, 3736
\g__xeCJK_punct_bound_width_tl 1607, 1905, 1907
\g__xeCJK_punct_style_seq 1560, 2151, 2157, 2160
\g__xeCJK_punct_width_tl 1606, 1805, 1807
\g__xeCJK_scale_family_prop 3178, 3219, 3232
\g__xeCJK_slant_factor_fp . . . 2281, 2297, 2301, 2388, 2409
\g__xeCJK_spacefactor_int 150, 163, 180, 198, 201, 203, 204, 664, 701
\g__xeCJK_special_punct_clist 1612, 1613, 1616, 1653
\g__xeCJK_sub_key_seq . . 2307, 2307, 2310, 2815, 2821, 2831
\g__xeCJK_under_symbol_box 4100, 4142, 4155
\g__xeCJK_unknown_family_seq 2710, 2712, 2718
\g__xunadd_encname_clist 4476, 4478, 4481, 4484, 4489, 4500, 4523, 4531
\g_fontspec_bfmathrm_tl 3442
\g_fontspec_encoding_tl 3265, 3367
\group_align_safe_begin: 242, 269, 1210, 1293, 1323, 1381, 1396
\group_align_safe_end: 239, 240, 268, 1213, 1214, 1295, 1297, 1326, 1334, 1340, 1384, 1399, 1400, 1406, 1413
\group_begin: 89, 207, 2262, 2420, 2469, 3185, 3426, 3628, 3716, 3728, 4112, 4448, 4647, 4738
\group_end: 89, 211, 216, 2275, 2435, 2475, 3185, 3426, 3629, 3721, 3732, 4120, 4451, 4651, 4743

H

\HalfLeft 363
\HalfRight 363
\hbar 4624, 4635, 4636, 4640
\hbox_overlap_right:n 4155
\hbox_set:Nn 4113, 4141, 4175
\hbox_set:Nw 208
\hbox_set_end: 211, 216
\hbox_to_zero:n 4145
\hbox_unpack:N 4187
\hcoffin_set:Nn 4804, 4805
CheckFullRight 5
CheckSingle 3
\hskip 3770, 3777

I

\icprotect 3675, 3676
MiddlePunct 4
MiddlePunct+ 4
MiddlePunct- 4
\if_case:w 223, 226, 3003
\if_catcode:w 253
\if_cs_exist:w 306, 1658
\if_dim:w 770
\if_false: 4439
\if_int_compare:w 231, 314, 1650
\if_meaning:w 247, 274, 3845
\if_predicate:w 4616
\IfBooleanF 2670
\IfBooleanT 479, 1456, 1466, 1475, 2680
\IfBooleanTF 1566, 2674, 4833, 4843
\ifCTEX@fntef 3691
\IfInstanceExistTF 2140, 2155, 2169
\IfNoValueF 3682
\IfNoValueTF 508, 509, 2770
\iftipaonetoken 4516, 4517, 4530
\ignorespacesafterend 4165, 4195
\indentfirst 3301
\InlineEnv 1418
\int_add:Nn 4290
\int_case:nnTF 1036
\int_compare:nNnF 3086, 3593, 4321

\int_compare:nNnTF 150, 163, 180, 209, 1075, 1078, 3007, 3011, 3239, 3559, 3935, 4077, 4422
\int_compare_p:nNn 635, 682, 730, 737, 746, 1043, 1064, 1065, 4338, 4339, 4359
\int_const:cn 346
\int_const:Nn 2971, 3105
\int_div_truncate:nn 200
\int_eval:n 290, 301, 310, 2786, 4678
\int_gincr:N 2401
\int_gset_eq:NN 204, 664, 701
\int_if_exist:cF 1500
\int_if_exist:cTF 332, 343, 1482
\int_if_odd:nTF 3098
\int_incr:N 517, 526, 2990
\int_max:nn 509
\int_min:nn 508
\int_new:N 54, 203, 234, 235, 2411, 2993, 4372
\int_set:cn 3605
\int_set:Nn 504, 508, 509, 512, 513, 522, 531, 3163
\int_set_eq:NN 505, 2998, 4328, 4349, 4369
\int_step_inline:nnnn 3596
\int_to_hexadecimal:n 2243
\int_use:N 286
\int_while_do:nNnn 1256
\int_zero:N 4136, 4381, 4392
\iow_indent:n 3489
\ItalicFont 2363
\IVS 363

J

CJKecglue 3
CJKglue 3
CJKmath 3
CJKspace 3

K

\KaiMingPunct 1583
\keys_define:nn 318, 792, 801, 830, 1181, 1225, 1269, 1418, 1449, 1555, 1583, 2128, 2198, 2282, 2311, 2363, 2368, 2947, 2994, 3282, 3301, 3307, 3369
\keys_set:nn 3337, 3398
\keys_set_known:nVN 2428
\keyval_parse:NNn 2460
\keyval_parse:NNV 2446
\KeyValue 1680, 1681, 1682, 1683, 1686, 1687, 1688, 1689

L

\l__kernel_expl_bool 4507, 4511
\l__keys_module_tl 3332
\l__peek_search_token 238, 254
\l__xeCJK_add_block_features_clist 2818, 2843, 2850, 2880
\l__xeCJK_add_font_features_clist 2814, 2816, 2840, 2849, 2879
\l__xeCJK_add_min_bound_to_margin_bool 1731, 1965
\l__xeCJK_auto_fake_bold_bool 2373, 2377, 2380, 2406, 2414, 2495
\l__xeCJK_auto_fake_slant_bool 2387, 2391, 2394, 2407, 2415, 2507
\l__xeCJK_begin_int 234, 512, 514, 516, 517, 521, 523, 525, 526, 2985, 2986, 2989, 2990
\l__xeCJK_bound_dim 1764, 1778, 1838, 1849, 1859, 1883, 1934, 1942, 1963, 1966
\l__xeCJK_bound_margin_ratio_fp 1729
\l__xeCJK_bound_margin_width_dim 1728
\l__xeCJK_bound_punct_ratio_fp 1727
\l__xeCJK_bound_punct_width_dim 1726
\l__xeCJK_ccglue_skip 797, 800, 3041, 3042, 3053, 3151, 3772, 3781, 3912, 4074
\l__xeCJK_check_single_cs_case_tl 1383, 1392, 1408, 1415
\l__xeCJK_CJK_group_bool 596, 598, 3235, 3507
\l__xeCJK_current_coor_tl 2599, 2601, 2608, 2610, 2613, 2630, 2642, 3177, 3179, 3220, 3258, 3261
\l__xeCJK_different_align_margin_dim 1741
\l__xeCJK_different_align_ratio_fp 1742
\l__xeCJK_ecglue_skip 739, 806, 823, 828, 3045, 3046, 3055, 3152, 3779, 3783, 3856
\l__xeCJK_embolden_factor_fp . 2374, 2381, 2408, 2416, 2498
\l__xeCJK_enabled_global_setting_bool 1713, 1800, 1900, 1974
\l__xeCJK_enabled_hanging_bool 1730, 1915
\l__xeCJK_enabled_kerning_bool 1734, 1979
\l__xeCJK_end_int 235, 513, 514, 521, 523, 2985, 2986
\l__xeCJK_env_cs_case_tl 1409, 1412, 1416
\l__xeCJK_env_cs_seq 1412
\l__xeCJK_fallback_family_tl 2223, 2224, 2233, 2236, 2242, 2263, 2265, 2269, 2270, 2272, 2277
\l__xeCJK_family_default_init_tl . 2894, 3384, 3386, 3395
\l__xeCJK_family_name_tl 2318, 2422, 2431, 2432, 2437, 2524, 2526, 2532, 2545, 2551, 2560, 2563
\l__xeCJK_fixed_margin_ratio_fp 1721
\l__xeCJK_fixed_margin_width_dim 1720
\l__xeCJK_fixed_punct_ratio_fp 1715
\l__xeCJK_fixed_punct_width_dim 1714
\l__xeCJK_fntef_shipout_tl 4094, 4125, 4131, 4138
\l__xeCJK_font_name_bf_tl 2365, 2403, 2493, 2503
\l__xeCJK_font_name_it_tl 2366, 2404, 2505, 2515
\l__xeCJK_font_name_tl 2265, 2266, 2337, 2338, 2424, 2438, 2473, 2524, 2812, 2845, 2875
\l__xeCJK_font_options_clist .. 2319, 2423, 2425, 2426, 2427, 2429, 2439, 2526, 2834, 2839, 2840, 2842, 2843, 2845
\l__xeCJK_font_options_prop . 2445, 2448, 2459, 2462, 2464
\l__xeCJK_fontspec_family_tl . 2471, 2474, 2478, 2563, 2567
\l__xeCJK_fontspec_options_clist 2405, 2413, 2429, 2472, 2497, 2502, 2509, 2514
\l__xeCJK_inline_env_case_tl 1398, 1444, 1446, 1448
\l__xeCJK_inline_env_seq 1422, 1429, 1430, 1437, 1441, 1445
\l__xeCJK_kerning_margin_minimum_dim 1745, 2018
\l__xeCJK_kerning_margin_ratio_fp 1744, 2061
\l__xeCJK_kerning_margin_width_dim ... 1743, 2059, 2060
\l__xeCJK_kerning_total_ratio_fp 1737, 2030, 2039
\l__xeCJK_kerning_total_width_dim 1736, 2027, 2028
\l__xeCJK_listings_env_bool 3088, 3106, 4210
\l__xeCJK_listings_flag_int 4321, 4328, 4338, 4339, 4349, 4359, 4369, 4372, 4381, 4392
\l__xeCJK_listings_letter_bool 4296, 4300, 4307, 4310, 4319, 4326, 4334, 4347, 4355, 4367, 4375, 4378, 4386, 4389
\l__xeCJK_margin_minimum_dim 1733, 1824, 1917
\l__xeCJK_middle_margin_ratio_fp 1725, 1857
\l__xeCJK_middle_margin_width_dim 1724, 1854, 1855
\l__xeCJK_middle_punct_ratio_fp 1719
\l__xeCJK_middle_punct_width_dim 1718
\l__xeCJK_min_bound_to_kerning_bool 1735, 2020
\l__xeCJK_mixed_margin_ratio_fp 1723
\l__xeCJK_mixed_margin_width_dim 1722
\l__xeCJK_mixed_punct_ratio_fp 1717
\l__xeCJK_mixed_punct_width_dim 1716
\l__xeCJK_new_line_cs_case_tl 1405, 1409, 1416
\l__xeCJK_new_line_cs_seq 1405
\l__xeCJK_no_break_cs_case_tl 1212, 1252
\l__xeCJK_no_break_cs_seq 1252
\l__xeCJK_off_verb_addon_tl 3065, 3080, 3108, 3109
\l__xeCJK_optimize_kerning_bool 1738, 2023
\l__xeCJK_optimize_margin_bool ... 1732, 1835, 1848, 1933
\l__xeCJK_peek_ignore_spaces_bool 241, 248, 265, 841, 846, 878, 882, 1314, 1327, 1335, 1347, 1352
\l__xeCJK_plain_equation_bool 1343, 1450
\l__xeCJK_punct_breakable_bool ... 1588, 1593, 1611, 4340
\l__xeCJK_reserve_space_bool 758, 832, 851, 889, 1316

\l__xeCJK_reset_shipout_skip_hook_tl .. 3026, 3077, 3083
\l__xeCJK_restore_listings_toks_tl 4207, 4215, 4216, 4223, 4233, 4242
\l__xeCJK_reverse_bound_dim 1766, 1779, 1838, 1842, 1859, 1892, 1928, 1966
\l__xeCJK_same_align_margin_dim 1739
\l__xeCJK_same_align_ratio_fp 1740
\l__xeCJK_scale_factor_fp 3213, 3215, 3218, 3231
\l__xeCJK_shipout_hook_bool 102, 104, 111
\l__xeCJK_shipout_hook_tl 105, 109, 110
\l__xeCJK_slant_factor_fp ... 2388, 2395, 2409, 2417, 2510
\l__xeCJK_sub_cancel_bool 1459, 1462, 1464, 1471, 1473, 1485
\l__xeCJK_sub_family_name_tl 2357, 2532, 2536, 2543, 2548, 2555, 2558, 2566
\l__xeCJK_sub_font_name_tl 2273, 2336, 2337, 2338, 2342, 2348, 2350, 2352, 2353, 2358, 2545, 2548, 2860, 2867, 2875, 2885
\l__xeCJK_sub_font_options_clist 2271, 2273, 2333, 2341, 2354, 2359, 2551, 2553, 2555, 2863, 2870, 2874, 2878, 2879, 2884
\l__xeCJK_sub_key_prop . 2317, 2323, 2339, 2402, 2412, 2530
\l__xeCJK_sub_key_seq 2817, 2823, 2828, 2837
\l__xeCJK_tmp_bool 57, 1255, 1258, 1260, 1267
\l__xeCJK_tmp_box 55, 208, 218, 1040, 1044, 1046, 1047, 4141, 4147, 4175, 4181, 4187
\l__xeCJK_tmp_clist 59, 483, 485, 486, 2972, 2974, 2975, 2976
\l__xeCJK_tmp_dim 56, 1798, 1821, 1826, 1831, 1842, 1872, 1881, 1883, 1891, 1892, 1898, 1913, 1919, 1923, 1928, 1940, 1942, 1970, 1986, 1996, 2102, 2110, 2118
\l__xeCJK_tmp_int ... 54, 352, 522, 525, 531, 533, 3163, 3165
\l__xeCJK_tmp_skip 58, 1261, 1263, 1267
\l__xeCJK_tmp_tl 53, 576, 577, 578, 582, 583, 585, 586, 2332, 2334, 2348, 2353, 2450, 2451, 2455, 2481, 2483, 2485, 2583, 2584, 2770, 2771, 2772, 2913, 2928, 2954, 2957, 2960, 2962, 4431, 4433
\l__xeCJK_ulem_hook_used_bool 3755, 3757, 3786, 4083, 4090, 4092, 4096
\l__xeCJK_ulem_skip_punct_bool ... 3759, 3789, 3791, 3792
\l__xeCJK_underdot_shipout_tl 4118, 4127, 4132, 4133
\l__xeCJK_verb_addon_bool 3112, 3114, 3167
\l__xeCJK_verb_case_int 2993, 2998, 3003, 3098
\l__xeCJK_verb_exspace_skip 3145, 3151, 3152, 3173, 3182, 3191, 3194, 3196, 3198, 3208
\l__xeCJK_xecglue_bool 634, 681, 736, 811, 816, 821, 829, 3068, 3069, 3132, 3133
\l__xunadd_add_circle_bool 4785, 4787, 4880, 4881
\l__xunadd_begin_hook_tl 4834, 4835, 4849, 4867
\l__xunadd_circle_coffin 4804, 4807, 4816, 4819, 4824
\l__xunadd_circle_ratio_fp 4807, 4826, 4827
\l__xunadd_circle_width_dim 4806, 4809, 4812, 4822
\l__xunadd_encname_clist 4503, 4504, 4509, 4533
\l__xunadd_end_hook_tl 4844, 4845, 4850, 4877
\l__xunadd_hyperref_hook_tl 4580, 4585, 4589
\l__xunadd_math_as_UTF_text_bool . 4552, 4556, 4562, 4566
\l__xunadd_math_as_UTF_text_seq 4554, 4559, 4560
\l__xunadd_tmp_coffin .. 4805, 4808, 4811, 4813, 4817, 4823
\l_keys_choice_int 2998
\l_keys_choice_tl 2285, 2293
\l_keys_key_tl 3326, 3327, 3371, 3372, 3373
\l_keys_value_tl 1558, 2140, 2141, 2142, 2289, 2297, 2381, 2395, 3295
\l_peek_token 247, 254, 274, 668, 705, 850, 851, 860, 888, 889, 1211, 1294, 1300, 1301, 1302, 1303, 1324, 1332, 1344, 1364, 1382
\l_xeCJK_current_font_tl 1663, 1665, 2592, 2592, 2593, 2597, 2599
\l_xeCJK_family_tl . 2223, 2239, 2247, 2248, 2593, 2600, 2628, 2631, 2632, 2634, 2638, 2641, 2671, 2678, 2679, 2691, 2692, 2696, 2696, 2782, 2783, 2812, 2834, 2860, 2863, 2874, 3171, 3174, 3177, 3179, 3181, 3207, 3220, 3226
\l_xeCJK_punct_style_tl 1663, 1665, 1752, 1772, 1784, 1790, 2137, 2141, 2145, 4206
\LA@space 3845
\labelsep 1059, 1067, 1068
PlainEquation 3
SlantFactor 4
\leavevmode 3506
AllowBreakBetweenPuncts 5
\LocalConfig 3282
\LongPunct 1583
\lst@AddToHook 4202, 4203, 4204, 4210
\lst@Append 4291, 4382, 4393
\lst@AppendLetter 4373, 4373
\lst@AppendOther 4373, 4384
\lst@arg 4433, 4444
\lst@FillOutputBox 4408
\lst@ifbreaklines 4217, 4381, 4392
\lst@ifec 4422
\lst@ifflexible 4397, 4404
\lst@ifletter 4299, 4312, 4325, 4346, 4366, 4380, 4391
\lst@IfNextCharActive 4428
\lst@InlineGJ 4441
\lst@InlineGJEnd 4446
\lst@InsideConvert@ 4435
\lst@lastother 4313, 4393
\lst@length 4290
\lst@letterfalse 4309, 4312, 4388, 4391
\lst@lettertrue 4297, 4299, 4322, 4325, 4343, 4346, 4363, 4366, 4377, 4380
\lst@numberstyle 4215
\lst@Output 4299, 4309, 4312, 4321, 4325, 4342, 4346, 4362, 4366, 4377, 4388, 4391
\lst@OutputOther 4299, 4325, 4346, 4366, 4380
\lst@ProcessLetter 4279
\lst@ProcessOther 4280
\lst@whitespacefalse 4295, 4306, 4318, 4333, 4354

M

\makexeCJKactive 321, 325, 325
\makexeCJKinactive 322, 325, 326, 3636, 3641, 3650, 3656, 3666
\math 3447, 3493
\math@s@text@true 4565
\mathrm 3435, 3443
EmboldenFactor 4
\mddefault 2964
\MiddlePunct 1583
\minipage 4160
\mode_if_math:TF 4576
\msg_critical:nn 10, 4473
\msg_critical:nnn 47
\msg_error:nn 62
\msg_error:nnn 30, 35, 50, 4528
\msg_error:nnx 63
\msg_info:nnxx 68
\msg_new:nnn 3, 17, 22, 38, 60
\msg_new:nnnn 61, 4467, 4534
\msg_redirect_module:nnn 3311, 3312, 3318, 3319
\msg_warning:nn 64
\msg_warning:nnx 65
\msg_warning:nnxx 66
\msg_warning:nnxxx 67

N

indentfirst 3
\newCJKfontfamily 6, 2766, 2768
\NewDocumentCommand 100, 325, 326, 476, 496, 535, 540, 547, 1253, 1452,

1460, 1469, 1564, 1578, 2153, 2167, 2258, 2666, 2745, 2751, 2756, 2766, 2768, 2774, 2800, 2803, 2948, 3084, 3107, 3396, 3401, 3403, 3405, 3406, 3407, 3409, 3410, 3411, 3413, 3415, 3417, 3419, 3680, 4708, 4825, 4829, 4839
\NewEnviron 4167
\NewLineCS 1404
\newXeTeXintercharclass 335
InlineEnv 4
InlineEnv+ 4
InlineEnv- 4
\nobreak 3357, 3506
\NoBreakCS 1251
\nobreakspace 3505, 3505, 3506
\normalfont 2748, 2754, 2759
\NormalSpace 363
\normalspacedchars 9, 535, 535
EnvCS 3
EnvCS+ 3
EnvCS- 3

O

NoBreakCS 5
NoBreakCS+ 5
NoBreakCS- 5
LocalConfig 2
LongPunct 4
LongPunct+ 4
LongPunct- 4
\or: 3004, 3006, 3010

P

\par 3355
\PassOptionsToPackage . . 3314, 3321, 3327, 3745, 3746, 4198
\pdfstringdefDisableCommands 3663, 4588, 4589
\pdftex_if_engine:T 4465
\pdftex_strcmp:D 223, 226
\peek_after:Nw 243, 249, 270, 275
\peek_catcode:NTF 1290, 1372
\peek_meaning_remove:NTF 665, 702
\penalty 3769, 3776
\Pifont 3655
\PlainEquation 1449
\prg_do_nothing: 128, 130, 561, 4495
\prg_new_conditional:Npnn 11, 143, 221, 304, 312, 768, 1648, 1656, 4591, 4612
\prg_new_protected_conditional:Npnn 2659
\prg_return_false: 14, 146, 227, 307, 316, 771, 1651, 1659, 2664, 4594, 4618, 4620
\prg_return_true: 14, 146, 224, 227, 307, 316, 771, 1651, 1659, 2664, 4594, 4618
\ProcessKeysOptions 3364
\ProcessOptions 3747, 4199, 4479
\prop_clear:N 2317, 2402, 2445
\prop_get:Nn 2255
\prop_get:NnN 2889
\prop_get:NnNT 2562, 2583
\prop_get:NnNTF 2781, 2890
\prop_get:NoN 2862
\prop_get:NoNTF 2859
\prop_get:NVN 2873
\prop_get:NVNF 2264
\prop_get:NVNT 2544, 2550, 2833, 2960
\prop_get:NVNTF 2811, 3178
\prop_get:NxN 2869
\prop_get:NxNTF 2866
\prop_gpop:NnNT 2481, 2483
\prop_gput:Nnn 2658, 2792
\prop_gput:NnV 2584
\prop_gput:NVV 2523, 2525, 2547, 2554, 3219
\prop_gput:Nxx 2650
\prop_if_empty:NF 2708
\prop_if_empty:NTF 2908
\prop_map_break:n 2923
\prop_map_inline:Nn 2448, 2530, 2921
\prop_new:N 2412, 2459, 2518, 2519, 2520, 2798, 3232
\prop_put:Nnn 2323, 2462, 2464
\prop_put:Nnx 2339
\protect 3453, 3454
\ProvideTextCommandDefault 4664
\PunctBoundWidth 1583
\PunctStyle 2128
\punctstyle 3405, 3405
\PunctWidth 1583

Q

\q_mark 2335, 2345
\q_nil 2335, 3818
\q_no_value 2323
\q_recursion_stop 3818
\q_recursion_tail 3818
\q_stop 285, 288, 300, 2335, 2345, 4697, 4857, 4861
\quark_if_nil:nTF 2347
\quark_if_no_value:nTF 2533
\quark_if_recursion_tail_stop:N 3839
\quiet 3307

R

\r 3642, 3643
\ReloadXunicode 4500, 4501, 4501
\RenewDocumentCommand 3506, 3542, 3655, 4081, 4110, 4501, 4550, 4563, 4596, 4603, 4622, 4684, 4722, 4729
\RenewDocumentEnvironment 4158
\RequirePackage 48, 51, 52, 3365, 3366, 3368, 3692, 3699, 3702, 3710, 3736, 3748, 3749, 3750, 3751, 4200, 4201, 4475, 4492, 4496
\reverse_if:N 231, 4380, 4397, 4404
\rmdefault 2900, 3547

S

\s__stop 119, 121
\scan_align_safe_stop: 3466, 3472
\scan_stop: 91, 208, 238, 860, 871, 872, 929, 936, 1860, 1953, 1963, 2043, 2064, 3235, 3463, 3507, 3747, 4135, 4199, 4479, 4517, 4593, 4700
\selectfont 2655, 3267, 3279
\seq_clear:N 2817
\seq_count:N 3594
\seq_gclear:c 1622
\seq_gclear:N 1560
\seq_gput_right:cn 1626, 1636
\seq_gput_right:Nn 337, 348, 471, 1536, 2310, 2712
\seq_gput_right:Nv 338
\seq_gput_right:Nx 2157
\seq_gremove_all:cn 1645
\seq_gset_eq:NN 464
\seq_gset_from_clist:Nn 466
\seq_if_empty_p:N 2828, 2837
\seq_if_in:cnF 1633
\seq_if_in:NnF 1235, 1429, 2710, 3598
\seq_if_in:NnTF 2821
\seq_map_function:NN 590, 2830
\seq_map_inline:cn 1620
\seq_map_inline:Nn 602, 901, 1248, 1445, 1511, 1527, 2815, 3606, 3819, 3821, 4231, 4256, 4270, 4554
\seq_new:c 1617
\seq_new:N 328, 329, 463, 465, 468, 1224, 1441, 1451, 2160, 2307, 2718, 4559
\seq_put_right:Nn 1235, 1430, 2823

\seq_remove_all:Nn 1240, 1437
\seq_set_from_clist:Nn 1422, 4560
\seq_set_split:Nnn 1229
\seq_use:Nnnn 2151
\setCJKfallbackfamilyfont 7, 7, 2258, 2258
\setCJKfamilyfont 5, 2732, 2766, 2766
\setCJKmainfont 5, 2728, 2745, 2745, 2750, 2761
\setCJKmathfont 7, 2762, 2948, 2948
\setCJKmonofont 5, 2730, 2745, 2756, 2764
\setCJKromanfont 2750, 2765
\setCJKsansfont 5, 2729, 2745, 2751, 2763
\SetSymbolFont 2968
\settextcircledratio 4825, 4825, 4828
\sfdefault 2901, 3548
\shapedefault 2964, 2966, 2969
\silent 3307
\skip_add:Nn 1263
\skip_horizontal:N 976, 1267, 3053, 3055, 3770, 3777, 3856, 3912
\skip_horizontal:n 951, 1067, 1068, 1171, 3057, 3339, 4071, 4078
\skip_if_eq:nnTF . 152, 161, 166, 183, 1059, 3042, 3046, 3145
\skip_if_eq_p:nn 636, 683, 738, 739, 1055
\skip_if_finite_p:n 1054
\skip_new:N 58, 800, 828, 3191
\skip_set:Nn 212, 217, 3152, 3173, 3194, 4079
\skip_set_eq:NN 1261, 3151, 4074
\skip_use:N 213, 3208
\skip_vertical:n 4144
\skip_zero:N 3182, 3198
\SlantFactor 2282
\SplitArgument 497
\str_case:nn 3553
\str_case:nnTF 2115, 3451
\str_case_x:nnn 2726, 2737, 2898
\str_case_x:nnTF 1403
\str_case_x:noTF 1397
\str_if_eq:nnF 488, 608, 2617
\str_if_eq:nnT 44, 3714
\str_if_eq:nnTF 604, 1515, 2322, 2620, 3823, 4624
\str_if_eq_x:nnTF 113, 2239, 2914, 3228, 3241, 3461
\str_if_eq_x_p:nn 1300, 3647, 3648
\sw@slant 3556

T

\tabular 4177, 4183, 4190
\tex_afterassignment:D 4693
\tex_baselineskip:D 3339
\tex_chardef:D 4700
\tex_font:D 141, 145, 154, 155, 156, 164, 170, 171, 176, 177, 186, 2108, 3091, 3092, 3187, 3195, 3202, 3239, 3260, 3268, 3275, 3428, 4593
\tex_fontdimen:D 154, 155, 156, 164, 170, 171, 176, 177, 186, 3091, 3195, 3202, 3260, 3268, 3275
\tex_futurelet:D 4393
\tex_global:D 2988
\tex_hfill:D 4161, 4169
\tex_hss:D 4148, 4409
\tex_iftrue:D 3691
\tex_ignorespaces:D 91, 914, 918, 2685, 2777, 2806, 3399, 4098, 4121
\tex_italiccorrection:D 665, 666, 702, 703
\tex_kern:D 781, 782, 941, 953, 4147
\tex_lastkern:D 770, 4077
\tex_lastpenalty:D 1065, 1075
\tex_lastskip:D 213, 636, 683, 738, 739, 1054, 1055, 1059, 1261, 1263
\tex_let:D 3676
\tex_noindent:D 4213
\tex_number:D 201, 4812, 4813
\tex_parindent:D 1044
\tex_penalty:D 69, 70, 1079, 1080, 3769, 3776
\tex_romannumeral:D 251, 277
\tex_space:D 3768, 3775
\tex_spacefactor:D 664, 701
\tex_spaceskip:D 152, 158, 161, 181, 186, 190, 191
\tex_textfont:D 3427
\tex_the:D 141, 559, 3428
\tex_unkern:D 785
\tex_unpenalty:D 1077
\tex_unskip:D 639, 686, 742, 1061, 1264
\tex_vrule:D 926, 933
\tex_vss:D 4150
\tex_xspaceskip:D 166, 174, 183, 188
\textaleph 4544
\textbeth 4545
\textbigcircle 4884
\textcentereddot 3632
\textcircled 4879, 4885, 4886, 4887, 4888, 4889, 4890, 4891, 4892, 4893, 4894, 4895, 4896, 4897, 4898, 4899, 4900, 4901, 4902, 4903, 4904, 4905, 4906, 4907, 4908, 4909, 4910, 4911, 4912, 4913, 4914, 4915, 4916, 4917, 4918, 4919, 4920, 4921, 4922, 4923, 4924, 4925, 4926, 4927, 4928, 4929, 4930, 4931, 4932, 4933, 4934, 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943, 4944, 4945, 4946, 4947, 4948, 4949, 4950, 4951, 4952, 4953, 4954, 4955, 4956, 4957, 4958, 4959, 4960, 4961, 4962, 4963, 4964, 4965, 4966, 4967, 4968, 4969, 4970, 4971, 4972, 4973, 4974, 4975, 4976, 4977, 4978, 4979, 4980, 4981, 4982, 4983, 4984, 4985, 4986, 4987, 4988
\textdaleth 4547
\textellipsis 3630, 3632
\textemdash 3632
\textFinv 4543
\textGame 4548
\textgimel 4546
\textperiodcentered 3632
\textquotedblleft 3633
\textquotedblright 3633
\textquoteleft 3633
\textquoteright 3633
\textvisiblespace 3243
\tipaencoding 3641
\tl_case:NnF 1393
\tl_case:NnTF 1218
\tl_case:NoF 1382
\tl_case:NoTF 1211
\tl_clear:N 1247, 1444, 2266, 2403, 2404, 4094, 4118
\tl_concat:NNN 1408, 1415
\tl_const:cn 36, 472
\tl_const:cx 1671, 1673, 3207
\tl_const:Nn 148, 1581, 1582, 2146, 2950, 3733, 3739
\tl_const:Nx 16, 2962, 3367
\tl_gput_right:Nn 75, 77, 79
\tl_gset:cn 1569, 1573, 1579
\tl_gset:Nn 2918, 3290, 3381, 3382, 3383, 3722, 3723
\tl_gset:Nx 999, 1012, 1030, 1096, 1106, 1113, 1137, 1146, 1155, 1177, 2896, 3295
\tl_gset_eq:NN 3395
\tl_gset_rescan:Nnn 2924
\tl_if_blank:nTF 2315, 4740, 4778, 4831, 4841
\tl_if_blank:VTF 2451, 2493, 2505
\tl_if_empty:NF 577, 583
\tl_if_empty:NT 3442
\tl_if_empty:NTF 1805, 1905, 2336, 2352
\tl_if_empty:nTF 292, 301
\tl_if_eq:NNT 2894
\tl_if_eq:NNTF 1752, 1784

\tl_if_exist:cF 1750, 1782
\tl_if_exist:cTF 13, 3171
\tl_if_exist:NF 3381, 3382, 3383
\tl_if_exist:NT 4480
\tl_if_exist:NTF 3390
\tl_if_exist_p:c 1975
\tl_if_head_eq_charcode:nNTF 4680
\tl_if_head_eq_meaning:nNTF 3463
\tl_if_head_is_N_type_p:n 123
\tl_if_single:nTF 4855
\tl_if_single_p:n 123
\tl_if_single_token:nTF 4614
\tl_if_single_token_p:n 1301
\tl_map_inline:nn 537, 1234, 1240, 1577, 2576, 3630
\tl_map_inline:xn 1568, 1572, 1623, 1631, 1642
\tl_new:c 1625, 1635
\tl_new:N 53, 71, 72, 73, 110, 263, 264, 954, 1223, 1392, 1448, 2145, 2277, 2357, 2358, 2437, 2438, 2478, 2592, 2613, 2696, 3083, 3109, 3299, 3384, 4131, 4132, 4242, 4585, 4849, 4850
\tl_put_left:Nn 3641, 4215
\tl_put_right:Nn 109, 1248, 1446, 2269, 2318, 3026, 3433, 4835, 4845
\tl_put_right:NV 4433
\tl_put_right:Nx 4233, 4580
\tl_replace_all:Nnn 585, 2361
\tl_replace_all:NnV 2338
\tl_replace_once:Nnn 3555, 3556, 3788, 3790
\tl_set:Nn 239, 240, 268, 2263, 2422, 2424, 2593, 4125, 4133, 4170, 4487, 4488, 4834, 4844
\tl_set:No 2450
\tl_set:Nx 115, 576, 582, 2141, 2233, 2247, 2332, 2350, 2532, 2599, 2608, 2630, 2654, 2678, 2691, 2770, 2954, 2957, 3065, 3177, 3258, 3266, 3386, 3544, 4223, 4522
\tl_set_eq:NN 2137, 2223, 2337, 2348, 2353, 2679, 2692, 2783, 2875, 2913, 3181, 4206
\tl_set_rescan:Nnn 4431, 4444
\tl_tail:N 3466
\tl_to_lowercase:n 3730, 4649, 4741
\tl_to_str:n 4600, 4607, 4611, 4629, 4688, 4695, 4712, 4718, 4726, 4733, 4758, 4761, 4862, 4868, 4874
\tl_trim_spaces:n 119, 125
\tl_use:c 1486, 1977
\tl_use:N 105, 3108, 4867, 4877
\token_get_arg_spec:N 1300, 3461
\token_get_replacement_spec:N 3451
\token_if_active:NTF 4416
\token_if_active_p:N 4617
\token_if_chardef:NTF 4768
\token_if_cs:NTF 1332, 4419
\token_if_cs_p:N 4617
\token_if_dim_register:NT 4171
\token_if_eq_meaning:NNTF 860, 871
\token_if_letter:NTF 4278, 4284
\token_if_letter_p:N 851, 889, 1303
\token_if_macro_p:N 850, 888
\token_if_math_toggle_p:N 1344
\token_if_other:NTF 1294, 1324
\token_if_other_p:N 1302
\token_if_space:NTF 668, 705
\token_to_meaning:N 285, 3476, 3482
\token_to_str:c 4611, 4695, 4718, 4746
\token_to_str:N 1495, 1548, 2728, 2729, 2730, 2732, 2739, 2740, 2741, 2854, 3224, 3229, 3378, 3379, 3476, 3482, 4538, 4611, 4635, 4636, 4654, 4661, 4662, 4664, 4695, 4718, 4746, 4856, 4862
\tonebar 4703, 4705, 4706
\TrimSpaces 476, 1453, 2153, 2167
\ttdefault 2902, 3228, 3229, 3241, 3549

U

\UL@hook 3752
\UL@leaders 4074, 4079
\UL@leadtype 3760, 3794, 3796
\UL@skip 4074, 4079
\UL@start 3872, 3888, 3905, 3925, 3938, 3952, 3974, 4002, 4025, 4040, 4062, 4070, 4090, 4097
\UL@stop 3872, 3888, 3905, 3921, 3933, 3947, 3961, 3996, 3999, 4021, 4035, 4062, 4070, 4084, 4092
\ULon 3788, 3789, 3790, 3791
\Umathcode 297
PunctBoundWidth *5*
PunctStyle *4*
PunctWidth *4*
\UndeclareTextCommand 3505, 4599, 4600
\UndeclareUTFcharacter 4596, 4596
\UndeclareUTFcomposite 4603, 4603
\Unicode 3717
\Url@MathSetup 3424, 3432, 3433
\use:c 475, 770, 781, 782, 1667, 1669, 1946, 1947, 1949, 1959, 1960, 2054, 2055, 2057, 2285, 2293, 2647, 2971, 3174, 4856, 4862
\use:n 1047, 1068, 1070, 1080, 1082, 1413, 1806, 1810, 1840, 1850, 1906, 1910, 1916, 1935, 2025, 2061, 2063, 2915, 3385, 3570, 3576, 4457
\use:x 2259, 2442, 2776, 3199, 3683, 4160, 4505
\use_i:nn 2452, 2455, 2537, 2662, 3848
\use_ii:nn 2450, 2537, 3846
\use_ii:nnn 1406
\use_iii:nnn 1384
\use_none:n 231, 232, 233, 1046, 1049, 1057, 1067, 1079, 1803, 1807, 1903, 1907, 2060, 3578, 4681, 4869, 4875
\use_none:nn 174, 188, 2470, 2670
\use_none:nnnnn 2958
\use_none_delimit_by_q_stop:w 4693
\usefont 3656
\UseInstance 1772, 1790
\UseMathAsText 4563, 4568
\usepackage 3378, 4538
\UTFencname 3505, 4480, 4481, 4484, 4487, 4488, 4489, 4522, 4525, 4526, 4596, 4603, 4622, 4684, 4708
AutoFakeBold *4, 5*
AutoFakeSlant *4, 5*
AutoFallBack *4*

V

\vbox_gset_to_ht:Nnn 4142
\Verb 2993
\verbatim@font 3018, 3019

X

\x@protect 3453
\xCJKecglue 801
\xe@alloc@intercharclass 3594, 3596
\xeCJK@family 2679, 2692, 2783, 3181
\xeCJK@fix@penalty 3554, 3554, 3555, 3556, 3808
\xeCJK@italiccorr 3555, 3557, 3557
\xeCJK@setfont 2614
\xeCJK_add_font_features:Nnn 2809, 2809, 2851
\xeCJK_add_font_features:Nnx 3217
\xeCJK_add_font_features:Nxx 2805
\xeCJK_allow_break: 69, 69
\xeCJK_app_inter_class_toks:nnn . . 568, 568, 573, 628, 914
\xeCJK_app_inter_class_toks:nnx 3610
\xeCJK_Boundary_and_Default: 627, 630, 630, 3806
\xeCJK_Boundary_and_FullLeft:N 911, 1022, 1022
\xeCJK_Boundary_and_FullRight:N 913, 1099, 1099
\xeCJK_Boundary_and_NormalSp: 676, 677, 677, 3807
\xeCJK_calc_punct_dimen:f 1763

\xeCJK_calc_punct_dimen:N 2096, 2096, 2122
\xeCJK_char_from_charcode:Nn 3719, 3726
\xeCJK_check_for_glue: 712, 718, 718, 3024, 3028, 3033, 3122, 3138, 3156
\xeCJK_check_FullRight: 1186, 1190, 1196, 1205, 1205
\xeCJK_check_FullRight_symbol:Nw 1191, 1219, 1219
\xeCJK_check_single:NNw 1291, 1295, 1305, 1310, 1310
\xeCJK_check_single:Nw 1274, 1277, 1282, 1288, 1288
\xeCJK_check_single_cs:NNn 1336, 1337, 1379, 1379
\xeCJK_check_single_env:nnNn 1385, 1394, 1394
\xeCJK_check_single_equation:NNnNw 1348, 1349, 1370, 1370
\xeCJK_CJK_and_Boundary:w 836, 837, 837, 3025, 3029, 3034, 3123, 3139, 3157
\xeCJK_CJK_and_CJK:N 895, 896, 896, 1274, 1276, 1277, 1282, 1283, 3805
\xeCJK_CJK_and_FullLeft:N 1008, 1008
\xeCJK_CJK_and_FullRight:N 1109, 1109
\xeCJK_class_group_begin: 593, 593, 617, 713, 995, 1027, 1092, 1103, 2222, 3896, 3906
\xeCJK_class_group_end: 593, 599, 623, 842, 843, 854, 857, 867, 970, 973, 987, 1216, 2225, 3034, 3160, 3871, 3887, 3904, 3946, 3982, 3996, 3999, 4020, 4034, 4061
\xeCJK_class_num:n 475, 475, 491, 494, 531, 538, 556, 559, 1650, 3163
\xeCJK_clear_Boundary_and_CJK_toks: 589, 589, 620, 714, 998, 1029, 1095, 1105, 3897, 3907
\xeCJK_clear_inter_class_toks:nn 560, 560, 592, 619, 997, 1094
\xeCJK_copy_inter_class_toks:nnnn 574, 574, 605, 607, 609, 1513, 1514, 1526, 1529, 1530, 3608, 3609, 3617, 3623
\xeCJK_cs_case_keys_define:nNNnn 1221, 1221, 1251, 1404, 1411
\xeCJK_cs_clear:N 127, 127, 3031, 3032, 3033, 3043, 3047, 3147, 3148, 3666, 3667, 3668, 3762, 3763, 3794, 3797, 4408
\xeCJK_cs_gclear:N 127, 129
\xeCJK_declare_char_class:nn 481, 481, 499, 1506
\xeCJK_declare_char_class:nV 542, 543, 544, 545, 549, 550, 551
\xeCJK_declare_char_class:nx 478, 1484
\xeCJK_declare_sub_char_class:nnn 1498, 1498, 1508
\xeCJK_declare_sub_char_class:nxx 1455
\xeCJK_def_node:nn ... 773, 773, 786, 787, 788, 789, 790, 791
\xeCJK_Default_and_FullLeft:nN 991, 991
\xeCJK_Default_and_FullRight:nN 1088, 1088
\xeCJK_fallback_loop:Nn 2224, 2228, 2228, 2236, 2248
\xeCJK_fallback_test_glyph:N . 2203, 2206, 2211, 2217, 2217
\xeCJK_family_if_exist:x 2659
\xeCJK_family_if_exist:xF 2632, 2911
\xeCJK_family_if_exist:xT 2560, 2574
\xeCJK_family_if_exist:xTF 2230, 2659, 2676, 2689, 2702, 2917, 2953, 2956
\xeCJK_family_if_exist_use:x 2671, 2674, 2700, 2700
\xeCJK_font_gset_to_current:c 138, 138, 2601, 2610, 2642, 3261
\xeCJK_fontspec:nn 2776, 2779, 2779, 2796
\xeCJK_fontspec:VV 2845
\xeCJK_FullLeft_and_CJK: 898, 955, 955, 3802
\xeCJK_FullLeft_and_Default: 965, 965, 3801, 4010
\xeCJK_FullLeft_and_FullLeft:N 1131, 1131
\xeCJK_FullLeft_and_FullRight:N 1140, 1140
\xeCJK_FullRight_and_Boundary: 914, 916, 917, 917, 1186, 1188, 1190, 1196, 1198
\xeCJK_FullRight_and_CJK: 900, 977, 977, 3804
\xeCJK_FullRight_and_Default: 918, 984, 984, 3803
\xeCJK_FullRight_and_FullLeft:N 1150, 1150
\xeCJK_FullRight_and_FullRight:N 1172, 1172
\xeCJK_FullRight_symbol:N 1097, 1107, 1114, 1148, 1179, 1189, 1191, 1199, 1204, 1204
\xeCJK_get_inter_class_toks:nn 558, 558, 565, 571, 576, 582, 3611, 3613, 3615, 3621, 4246
\xeCJK_get_punct_bounds:NN ... 993, 1010, 1024, 1090, 1101, 1111, 1134, 1143, 1152, 1174, 1207, 1748, 1748, 3994
\xeCJK_get_punct_kerning:NN 1780, 1780, 1795
\xeCJK_get_punct_kerning:oN 1135, 1144, 1153, 1175
\xeCJK_glue_to_skip:nN 205, 205, 797, 806, 823, 3041, 3045, 3766, 3773
\xeCJK_glyph_bounds:NN 2099, 2101, 2123, 2123
\xeCJK_glyph_if_exist:N 143
\xeCJK_glyph_if_exist:NTF 143, 2219, 2234, 3236
\xeCJK_glyph_if_exist_p:N 143
\xeCJK_gset_mathcode:nnnn 2978, 2983, 2983
\xeCJK_hook_for_ulem: 3752, 3753, 3753
\xeCJK_if_blank_x:n 221
\xeCJK_if_blank_x:nT 3614, 3620
\xeCJK_if_blank_x:nTF 221, 504, 2668
\xeCJK_if_blank_x_p:n 221, 502
\xeCJK_if_CJK_class:N 304
\xeCJK_if_CJK_class:NTF 304, 1362, 1364
\xeCJK_if_CJK_class_p:N 304
\xeCJK_if_last_node:n 768
\xeCJK_if_last_node:nT 655, 695, 884, 3857, 3864
\xeCJK_if_last_node:nTF .. 724, 768, 3561, 3567, 3573, 3855
\xeCJK_if_last_node_p:n 642, 643, 650, 651, 689, 690, 721, 729, 745, 747, 753, 754, 768
\xeCJK_if_package_loaded:n 11
\xeCJK_if_package_loaded:nF 3313, 3320, 3696
\xeCJK_if_package_loaded:nT 29, 3530, 3653, 3661, 3701, 3709
\xeCJK_if_package_loaded:nTF 11, 34, 80, 3325, 3514, 3520, 3698
\xeCJK_if_package_loaded_p:n 11, 3498, 3499, 3675
\xeCJK_if_same_class:NN 312
\xeCJK_if_same_class:NNTF 312, 2032
\xeCJK_if_same_class_p:NN 312
\xeCJK_if_ulem_patch:TF 3843, 3843, 3853, 3863, 3869, 3879, 3885, 3902, 3919, 3931, 3944, 3959, 3980, 3990, 4008, 4017, 4031, 4047, 4053, 4059, 4069, 4103
\xeCJK_ignore_spaces:w 874, 874, 3585
\xeCJK_int_until_do:nn 230, 230, 514, 523, 2986
\xeCJK_inter_class_toks:nnc 907, 922
\xeCJK_inter_class_toks:nnn 555, 555, 557, 561, 615, 623, 627, 662, 675, 699, 710, 836, 895, 897, 899, 910, 912, 915, 3825, 3827, 3831, 4236, 4238, 4245, 4250, 4252, 4254, 4258, 4264, 4266, 4268, 4272
\xeCJK_inter_class_toks:nnx 564, 570, 578, 586, 905
\xeCJK_make_node:n ... 666, 669, 670, 703, 706, 707, 773, 779, 854, 857, 867, 885, 1375, 1388, 3564, 3570, 3576, 3873
\xeCJK_new_class:n 330, 330, 363, 364, 365, 366, 1502
\xeCJK_new_sub_key:n 1504, 2307, 2308, 2362
\xeCJK_no_break: 69, 70, 725, 960, 963, 970, 973, 1123, 1127, 1133, 1142, 1147, 1161, 1178, 1213, 1266, 1374, 1387, 3053, 3055, 3057, 3968, 3972, 3996, 4001, 4156
\xeCJK_peek_after_ignore_spaces:nw 266, 266, 1220
\xeCJK_peek_catcode_ignore_spaces:NTF 236, 236, 839, 876, 1312
\xeCJK_pre_inter_class_toks:nnn 562, 562, 567, 673, 1517, 1531, 1533, 1540
\xeCJK_pre_inter_class_toks:nnx 3612
\xeCJK_punct_kern:NN 1154, 1163, 1589, 1594
\xeCJK_punct_kerning_process:NN 1791, 1968, 1968
\xeCJK_punct_margin_process:NN 1773, 1796, 1796
\xeCJK_punct_offset_process:NN 1774, 1896, 1896
\xeCJK_remove_node: .. 645, 653, 656, 692, 696, 722, 725, 732, 749, 757, 784, 885, 3563, 3569, 3575, 3856, 3857, 3864
\xeCJK_replace_inter_class_toks:nnnn 580, 580, 1521
\xeCJK_reverse:nnn 112, 112, 1769
\xeCJK_save_class:nn 341, 341, 358, 359, 360, 361, 362

\xeCJK_select_font: 618, 715, 996, 1028, 1093, 1104, 1763, 2592, 2595, 2614, 2621, 3185, 3426, 3898
\xeCJK_select_font:x 2232, 2592, 2604
\xeCJK_set_char_class:nnn 490, 494, 519, 519, 3421
\xeCJK_set_family:nnn 2418, 2418, 2440, 2442, 2793
\xeCJK_set_family:Voo 2536
\xeCJK_set_family:VVV 2272
\xeCJK_set_family:xxx 2441, 2747, 2753, 2758, 2767, 2772, 2949
\xeCJK_set_family_fallback:nnn 2259, 2260, 2260
\xeCJK_set_mathfont: 2931, 2951, 2951
\xeCJK_set_visible_space_font: 3253, 3256, 3256
\xeCJK_space_or_xecglue: 656, 696, 812, 817, 824, 842, 855, 879, 891
\xeCJK_swap_cs:NN 131, 131, 3840, 4105, 4453
\xeCJK_switch_family:n 2687, 2771, 2794, 2930, 3547, 3548, 3549, 3550
\xeCJK_tl_remove_outer_braces:N 114, 114
\xeCJK_tl_remove_outer_braces:n 114, 115, 116, 124, 2332, 2351
\xeCJK_token_value_charcode:N .. 283, 284, 284, 2108, 2126
\xeCJK_token_value_class:N 282, 282, 306, 315, 1650
\xeCJK_visible_space: 3233, 3233, 3249
\xeCJK_visible_space_fallback: 3245, 3250, 3250
\xeCJK_xetex_mathcode:w 297, 302, 2988
\xeCJKactive 318
\xeCJKallowbreakbetweenpuncts 3411, 3411
\xeCJKCancelSubCJKBlock 9, 1459, 1460
\xeCJKcaption 3678, 3680
\xeCJKDeclareCharClass 9, 476, 476
\xeCJKDeclarePunctStyle 10, 2153, 2153, 2166, 2174, 2175, 2176, 2183, 2191
\xeCJKDeclareSubCJKBlock 8, 1452, 1452, 1458, 1495
\xeCJKdisablefallback 3415, 3417
\xeCJKEditPunctStyle 10, 2167, 2167, 2173
\xeCJKenablefallback 3415, 3415
\xeCJKnobreak 12, 1253, 1253
\xeCJKnobreakbetweenpuncts 3411, 3413
\xeCJKOffVerbAddon 12, 3084, 3107, 3124, 3142
\xeCJKplainchr 3405, 3406
\xeCJKResetCharClass 9, 547, 547, 554
\xeCJKResetPunctClass 9, 479, 540, 540, 552, 1456, 1466, 1475, 3422
\xeCJKRestoreSubCJKBlock 9, 1459, 1469
\xeCJKsetcharclass 3419, 3419
\xeCJKsetecglue 3408
\xeCJKsetemboldenfactor 3401, 3401
\xeCJKsetkern 10, 1578, 1578, 1580
\xeCJKsetslantfactor 3401, 3403
\xeCJKsetup 2, 1561, 3081, 3143, 3379, 3396, 3396, 3402, 3404, 3405, 3406, 3407, 3409, 3410, 3412, 3414, 3416, 3418, 3758
\xeCJKsetwidth 9, 1564, 1564, 1576
\xeCJKShipoutHook 12, 99, 99, 100
\xeCJKVerbAddon 12, 3008, 3012, 3084, 3084, 3168, 3224
\xetex_if_engine:F 10
\xetex_if_engine:TF 4486, 4491
\xetex_XeTeXversion:D 286
\XeTeXcharclass 283, 525, 533, 538, 3165
\XeTeXcharglyph 2126
\XeTeXdefaultencoding 3682, 3689
\XeTeXfonttype 3239
\XeTeXglyphbounds 1544, 1548, 2125
\XeTeXinterchartokenstate 325, 326, 3559
\XeTeXinterchartoks 556, 559
\XeTeXmathcode 302
\XeTeXpicfile 4494, 4495, 4497
\XeTeXrevision 286