xeCJK 宏包

ctex.org

2012/12/13　　3.1.1*

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 . . 9
- 3.6 其它 11

4 已知问题和兼容性 11

5 xeCJK 代码实现 12
- 5.1 运行环境检查 12
- 5.2 内部工具 12
- 5.3 功能开关 16
- 5.4 字符类别设定 17
- 5.5 字符类别处理 20
- 5.6 字符输出规则 23
- 5.7 全角右标点后的断行 32
- 5.8 段末孤字处理 33
- 5.9 增加 CJK 子分区 37
- 5.10 标点处理 38
- 5.11 后备字体 48
- 5.12 CJK 字体族声明方式 50
- 5.13 字体切换 57
- 5.14 数学字体设置 62
- 5.15 抄录环境中的间距调整 63
- 5.16 xeCJK 其它选项 65
- 5.17 xeCJK 初始化设置 66
- 5.18 兼容性修补 68
- 5.19 xeCJKfntef 72
- 5.20 xeCJK.cfg 78

版本历史 79

代码索引 79

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 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 `CJKnumber` 需要 CJK 宏包套件中的 `CJKnumb` 的支持；`indentfirst` 选项需要 indentfirst 宏包。xeCJK 会自动根据需要载入这些宏包。

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

*ctex-kit rev382.

[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` {⟨key_1⟩=⟨var_1⟩, ⟨key_2⟩=⟨var_2⟩, ...}

其中 ⟨key_1⟩, ⟨key_2⟩ 是设置选项，而 ⟨val_1⟩, ⟨val_2⟩ 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 TeXLive 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。

CJKnumber ★

CJKnumber = ⟨*true*|***false***⟩

为了与 CJKnumb 宏包兼容，xeCJK 作了一些特别处理。因此，如果需要使用到 CJKnumb，请通过这个选项调用它。

indentfirst ★

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、CJKnumber 和 indentfirst 这三个选项需要在调用 xeCJK 时设置，而不能通过 \xeCJKsetup 来设置。

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。

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: 2012-12-06

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 = {⟨*env*$_1$⟩, ⟨*env*$_2$⟩, ⟨*env*$_3$⟩, ...}

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；
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。

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，可能出现了意料之外的断行。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。

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。

\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

\addCJKfontfeatures {⟨font features⟩}

临时增加当前使用的 CJK 字体的选项。例如，使用

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

可以将文档中当前使用的 CJK 字体放大为 1.1。

\CJKrmdefault

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

\CJKfamilydefault

保存 \normalfont 所使用的 CJK 字体族，默认值是 \CJKrmdefault。类似西文字体的 \familydefault。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。例如，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 ★

```
\setCJKfallbackfamilyfont {⟨family⟩} [⟨common font features⟩]
  {
    {[⟨font features 1⟩] {⟨font name 1⟩}} ,
    {[⟨font features 2⟩] {⟨font name 2⟩}} ,
     ......
  }
```

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

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

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

3.2.1 XƎTEX 的字体名查找

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

XƎTEX 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:

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

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

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

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

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

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

3.3 CJK 分区字体设置

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

\xeCJKDeclareSubCJKBlock ⋆

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

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

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

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

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

就声明了 SPUA 和 Ext-B 这两个个子分区。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  {⟨block1,block2,...⟩}
\xeCJKCancelSubCJKBlock* {⟨block1,block2,...⟩}
```

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

\xeCJKRestoreSubCJKBlock

```
\xeCJKRestoreSubCJKBlock  {⟨block1,block2,...⟩}
\xeCJKRestoreSubCJKBlock* {⟨block1,block2,...⟩}
```

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

3.4 设置 CJK 字符范围

\xeCJKDeclareCharClass ★

\xeCJKDeclareCharClass {⟨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 ★

\xeCJKResetCharClass

用于恢复 xeCJK 对各个字符类别的初始化设置。

\xeCJKResetPunctClass ★

\xeCJKResetPunctClass

用于重置标点符号所属的字符类。

\normalspacedchars

\normalspacedchars {⟨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。

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 plain 以外的所有标点处理格式。

\xeCJKsetwidth ★

\xeCJKsetwidth {⟨标点列表⟩} {⟨length⟩}

⟨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。

\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 ★

New: 2012-11-10

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

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

\xeCJKEditPunctStyle ★

New: 2012-11-10

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

修改已有的标点符号处理格式。

下面是可以设置的标点符号格式选项。其中左边一栏是选项名称,中间是选项的输入值类型,右边则是相关说明。某些选项之间是互斥的,具有优先级关系。要使下一级的选项有效,则需要先禁用上一级的设置:对于 ⟨*boolean*⟩ 类型的选项,将其设置为 false,对于 ⟨*length*⟩ 类型的选项,将其设置为 \maxdimen,而对于 ⟨*real*⟩ 类型的选项,将其设置为 0。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

就定义了一个名为 mine 的标点处理格式。可以在通过

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

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

```
\xeCJKEditPunctStyle { 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

New: 2012-11-19

\xeCJKVerbAddon

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

\xeCJKnobreak

New: 2012-12-03

……汉字。\xeCJKnobreak\footnote{脚注}

\xeCJKnobreak 用在全角标点符号后面，目的是确保不能在此处断行。如果已经启用了前面介绍的 CheckFullRight 选项，则不需要再用此命令。

4 已知问题和兼容性

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

xeCJK 包含有一个子宏包 xeCJKfntef，可以用它来实现汉字加点和可断行的下划线等。它是 CJKfntef 宏包在在 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。

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(例如 peek 类函数)在遇到 \outer 宏时可能会存在问题。按照目前 xeCJK 的实现方式，在 CJK 文字后面遇到 \outer 宏时会出现类似

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

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

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

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

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

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

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

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

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 } }
```

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 } { 2012/11/10 } { }
  { \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 }
```

5.2 内部工具

分配临时变量。

```
\tl_new:N \l__xeCJK_tmpa_tl
\tl_new:N \l__xeCJK_tmpb_tl
\int_new:N \l__xeCJK_tmpa_int
\int_new:N \l__xeCJK_tmpb_int
\int_new:N \l__xeCJK_tmpc_int
\box_new:N \l__xeCJK_tmp_box
\dim_new:N \l__xeCJK_tmp_dim
\skip_new:N \l__xeCJK_tmp_skip
\prop_new:N \l__xeCJK_tmp_prop
\clist_new:N \l__xeCJK_tmpa_clist
\clist_new:N \l__xeCJK_tmpb_clist
```

__xeCJK_msg_new:nn
__xeCJK_error:n
__xeCJK_error:nx
__xeCJK_warning:nx
__xeCJK_info: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_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_info:nx     { \msg_info:nnx     { 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_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)

__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)

\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
  { \__xeCJK_tl_remove_outer_braces:w #1 \q_stop }
\cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \q_stop
  {
    \bool_if:nTF { \tl_if_single_p:n {#1} && \tl_if_head_is_group_p:n {#1} }
      { \__xeCJK_tl_remove_outer_braces:w #1 \q_stop }
      { \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_eq:NN \xeCJK_cs_clear:N \tl_clear:N
\cs_new_eq:NN \xeCJK_cs_gclear:N \tl_gclear:N
```

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

\xeCJK_swap_cs:NN

交换 #1 和 #2 的意义。

```
\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。没有考虑到 \xspaceskip 和 \spacefactor 的情况。

```
\tl_const:Nn \c_xeCJK_space_skip_tl
  {
    \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 }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_glue_to_skip:nN #1#2
  {
    \hbox_set:Nn \l__xeCJK_tmp_box { #1 \scan_stop: \exp_after:wN } \exp_after:wN
    \skip_set:Nn \exp_after:wN #2 \exp_after:wN { \skip_use:N \tex_lastskip:D }
  }
```

(*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} }
```

(*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:
  }
\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 }
\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 }
\cs_new_nopar:Npn \__xeCJK_token_value_charcode:w #1 ~ #2 ~ #3 \q_stop { `#3 }
```

(*End definition for* \xeCJK_token_value_class:N)

\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_tmpa_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。

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

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`

新增西文半角左/右标点和前后原始间距的符号类。

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

(*End definition for* `HalfLeft`, `HalfRight`, *and* `NormalSpace`)

`\c__xeCJK_HalfLeft_chars_clist`
`\c__xeCJK_HalfRight_chars_clist`
`\c__xeCJK_NormalSpace_chars_clist`
`\c__xeCJK_FullLeft_chars_clist`
`\c__xeCJK_FullRight_chars_clist`
`\c__xeCJK_CJK_chars_clist`

xeCJK 设置的基本字符类的默认范围。

```
\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 }
\clist_const:Nn \c__xeCJK_FullLeft_chars_clist
  {
     "2018 , "201C , "2116 , "3008 , "300A , "300C , "300E , "3010 , "3012 ,
     "3014 , "3016 , "3018 , "301A , "301D , "3036 , "E76C , "FE59 , "FE5B ,
     "FE5D , "FE5F , "FE60 , "FE69 , "FE6B , "FF03 , "FF04 , "FF08 , "FF20 ,
     "FF3B , "FF5B , "FFE0 , "FFE1 , "FFE5 , "FFE6
  }
\clist_const:Nn \c__xeCJK_FullRight_chars_clist
  {
     "00B7 , "2019 , "201D , "2014 , "2015 , "2025 , "2026 , "2030 , "2500 ,
     "3001 , "3002 , "3005 , "3006 , "3009 , "300B , "300D , "300F , "3011 ,
     "3015 , "3017 , "3019 , "301B , "301E , "301F , "3041 , "3043 , "3045 ,
     "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E , "309B , "309C ,
     "309D , "309E , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 ,
     "30E5 , "30E7 , "30EE , "30F5 , "30F6 , "30FB , "30FC , "30FD , "30FE ,
     "FE50 , "FE51 , "FE52 , "FE54 , "FE55 , "FE56 , "FE57 , "FE5A , "FE5C ,
     "FE5E , "FE6A , "FF01 , "FF05 , "FF09 , "FF0C , "FF0E , "FF1A , "FF1B ,
     "FF1F , "FF3D , "FF5D , "FF61 , "FF63 , "FF64 , "FF65 , "FF67 , "FF68 ,
     "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F , "FF70 , "FF9E ,
     "FF9F
  }
\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 ,
```

- Kana Supplement (日文假名增补)

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

- 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 ,
```

- 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 ,
```

- 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_HalfLeft_chars_clist` *and others.*)

5.5 字符类别处理

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

```
\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
\prop_new:N \g__xeCJK_CJK_class_prop
\cs_new_protected_nopar:Npn \__xeCJK_save_CJK_class:n #1
  {
    \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
    \prop_gput:Nvn \g__xeCJK_CJK_class_prop { \__xeCJK_class_csname:n {#1} } {#1}
    \tl_gput_right:Nx \g__xeCJK_check_single_CJK_case_tl
      { { \exp_not:c { \__xeCJK_class_csname:n {#1} } } { \exp_not:N \use_ii:nn } }
  }
\tl_new:N \g__xeCJK_check_single_CJK_case_tl
\cs_generate_variant:Nn \prop_gput:Nnn { Nvn }
\clist_map_function:nN { CJK , FullLeft , FullRight } \__xeCJK_save_CJK_class:n
```

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

`\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_tmpa_clist {#2}
    \clist_gconcat:ccN
      { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } \l__xeCJK_tmpa_clist
    \clist_map_inline:Nn \l__xeCJK_tmpa_clist
      {
        \str_if_eq:nnF {##1} { -> }
          {
```

```
            \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
              { \xeCJK_class_num:n {#1} }
          }
      }
  }
\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_set_char_class_catcode:nnn

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

```
\cs_new_protected_nopar:Npn \__xeCJK_set_char_class_catcode:nnn #1#2#3
  {
    \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
    \int_set:Nn \l__xeCJK_tmpc_int {#3}
    \int_compare:nNnTF
      { \use:c { \__xeCJK_class_csname:n { CJK } } } = \l__xeCJK_tmpc_int
      { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \char_set_catcode_other:n }
      { \cs_set_eq:NN \__xeCJK_set_char_catcode:n \use_none:n }
    \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
      {
        \__xeCJK_set_char_catcode:n { \l__xeCJK_tmpa_int }
        \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
        \int_incr:N \l__xeCJK_tmpa_int
      }
  }
```

(*End definition for* __xeCJK_set_char_class_catcode:nnn)

__xeCJK_set_char_class:nnn

只设置字符类别,而不改变 \catcode。

```
\cs_new_protected_nopar:Npn \__xeCJK_set_char_class:nnn #1#2#3
  {
    \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_tmpa_int \l__xeCJK_tmpb_int
    \int_set:Nn \l__xeCJK_tmpc_int {#3}
    \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
      {
        \XeTeXcharclass \l__xeCJK_tmpa_int = \l__xeCJK_tmpc_int
        \int_incr:N \l__xeCJK_tmpa_int
      }
  }
```

(*End definition for* __xeCJK_set_char_class:nnn)

__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)

\xeCJK_set_char_class:nnn

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

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

(*End definition for* \xeCJK_set_char_class:nnn)

__xeCJK_set_char_class_eq:nn

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

```
\cs_new_protected_nopar:Npn \__xeCJK_set_char_class_eq:nn #1#2
  {
    \int_set:Nn \l__xeCJK_tmpa_int { \xeCJK_class_num:n {#2} }
    \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
      { \XeTeXcharclass ##1 = \l__xeCJK_tmpa_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
    \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 , nnv , nnx }
```

(*End definition for* \xeCJK_inter_class_toks:nnn)

\xeCJK_get_inter_class_toks:nn

取出相邻类别之间的内容。

```
\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
  {
    \xeCJK_inter_class_toks:nnx {#1} {#2}
      { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
  }
```

(*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_tmpa_tl { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
    \tl_replace_all:Nnn \l__xeCJK_tmpa_tl {#3} {#4}
    \xeCJK_inter_class_toks:nnx {#1} {#2} { \exp_not:V \l__xeCJK_tmpa_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
Default		✓	✓	✓				✓
CJK	✓	✓	✓	✓	✓	✓	✓	✓
FullLeft	✓	✓	✓	✓	✓	✓	✓	✓
FullRight	✓	✓	✓	✓	✓	✓	✓	✓
HalfLeft		✓	✓	✓				
HalfRight		✓	✓	✓				✓
NormalSpace		✓	✓	✓				✓
Boundary	✓	✓	✓	✓	✓		✓	

\xeCJK_class_group_begin:
\xeCJK_class_group_end:

```
\cs_new_eq:NN \xeCJK_class_group_begin: \c_group_begin_token
\cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token
```

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

```
\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 , NormalSpace }
  {
    \xeCJK_inter_class_toks:nnn { Boundary } {#1}
      {
        \bool_if:nTF
          {
            \l__xeCJK_xecglue_bool &&
            \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 }
              }
              { \CJKecglue } { ~ }
          }
          {
            \xeCJK_if_last_node:nTF { CJK } { \CJKecglue }
              { \xeCJK_if_last_node:nT { CJK-space } { \__xeCJK_space_or_xecglue: } }
          }
      }
    \str_if_eq:nnF {#1} { NormalSpace }
      { \xeCJK_app_inter_class_toks:nnn { CJK } {#1} { \CJKecglue } }
  }
\clist_map_inline:nn { Default , HalfRight , NormalSpace }
  {
    \xeCJK_inter_class_toks:nnn {#1} { Boundary }
      {
        \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 } } }
          }
      }
    \str_if_eq:nnF {#1} { NormalSpace }
      { \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue } }
  }
\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 } }
      { \CJKglue }
      {
        \bool_if:nTF
          {
            \xeCJK_if_last_node_p:n { default }              ||
            \int_compare_p:nNn \etex_lastnodetype:D = \c_ten
          }
          { \CJKecglue }
          {
            \bool_if:nT
              {
                \l__xeCJK_xecglue_bool &&
                ( \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 }
                  }
                  { \CJKecglue }
                  {
                    \bool_if:nTF
                      {
                        \xeCJK_if_last_node_p:n { CJK }       ||
                        \xeCJK_if_last_node_p:n { CJK-space }
                      }
                      { \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 }
  }
\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 }
```

(*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
xCJKecglue CJK 与西文和数学行内数学公式之间自动添加的空白。

```
\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_class_group_end:
    { \xeCJK_make_node:n { CJK } }
    \xeCJK_ignore_spaces: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:NTF \l__xeCJK_peek_ignore_spaces_bool
          {
            \tex_unkern:D \tex_unkern:D
            { \xeCJK_make_node:n { CJK-space } }
            \bool_if:nT
              {
                \token_if_macro_p:N \l_peek_token ||
                ( \l__xeCJK_reserve_space_bool &&
                  \token_if_other_p:N \l_peek_token )
              }
              { \__xeCJK_space_or_xecglue: }
          }
          { \cs_if_exist:NF \l_peek_token \exp_not:N }
      }
  }
```

(*End definition for* \xeCJK_ignore_spaces:w)

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

\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_rule:NN 用于抹去标点符号的左/右空白。

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

(*End definition for* __xeCJK_punct_rule:NN)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_punct_glue:NN #1#2
  {
    \__xeCJK_punct_hskip:n
      {
        \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2}
        minus \dim_eval:n { ( \__xeCJK_use_punct_dim:nnn { glue } {#1} {#2} ) / \c_two }
      }
  }
\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} }
```

(*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_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_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: }
  }
```

(*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_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_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} }
```

(*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} }
```

(*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_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}
  }
\cs_new_protected_nopar:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
  {
    \int_compare:nNnF \etex_lastnodetype:D = \c_one
      { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
  }
```

(*End definition for* `\xeCJK_Boundary_and_FullLeft: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_punct_if_middle:NT {#2}
      { \__xeCJK_punct_rule:NN \c__xeCJK_left_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_punct_if_middle:NT {#1}
      { \__xeCJK_punct_rule:NN \c__xeCJK_left_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_punct_if_middle:NT {#1}
      { \__xeCJK_punct_rule:NN \c__xeCJK_left_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_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}
    \__xeCJK_punct_break:
    \CJKpunctsymbol {#1}
  }
```

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

`\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
    \tl_case:Non \l_peek_token { \l__xeCJK_no_break_cs_case_tl } {  }
    \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \xeCJK_class_group_end:
  }
\cs_generate_variant:Nn \tl_case:Nnn { 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 #2 {####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 { \xeCJK_no_break: } { }
```

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

\xeCJKnobreak 为保险起见,我们在这里用了一个循环。

```
\NewDocumentCommand \xeCJKnobreak { }
  {
    \int_zero:N \l__xeCJK_tmpa_int
    \int_while_do:nNnn \etex_lastnodetype:D = \c_eleven
      {
        \int_if_even:nTF \l__xeCJK_tmpa_int
          {
            \int_incr:N \l__xeCJK_tmpa_int
            \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:
    \int_if_even:nF \l__xeCJK_tmpa_int
      { \skip_horizontal:N \l__xeCJK_tmp_skip }
  }
```

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

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```
\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_other_token
      { \xeCJK_check_single:NNw #1 }
      {
        \bool_if:nTF
          {
            \xeCJK_if_blank_x_p:n { \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 \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_other_token
      {
        \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
          { \__xeCJK_check_single_space:NN #1#2 }
          { \__xeCJK_check_single_save:N #1 #2 }
      }
      {
        \group_align_safe_begin:
        \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
  {
    \int_case:non { \xeCJK_token_value_class:N #2 }
      { \g__xeCJK_check_single_CJK_case_tl }
      { \use_i:nn }
      { \__xeCJK_check_single_save:N #1 #2 ~ }
      {
        \int_case:non { \xeCJK_token_value_class:N \l_peek_token }
          { \g__xeCJK_check_single_CJK_case_tl }
          { \use_i:nn }
          { \__xeCJK_check_single_save:N #1 #2 ~ }
          { \__xeCJK_check_single_save:N #1 #2 }
      }
  }
\cs_generate_variant:Nn \int_case:nnn { no }
```

(*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_no_break: #2 #4
      }
      { \__xeCJK_check_single_save:N #1 #2#3#4 }
  }
```

(*End definition for* \xeCJK_check_single_equation:NNnNw)

\xeCJK_check_single_cs:NNn

```
\cs_new_protected_nopar:Npn \xeCJK_check_single_cs:NNn #1#2#3
  {
    \tl_case:Non \l_peek_token
      { \l__xeCJK_check_single_cs_case_tl }
      { \use_iii:nnn }
      { \xeCJK_check_single_env:nnNn }
      {
        \xeCJK_no_break: \__xeCJK_check_single_save:N #1
        \xeCJK_no_break: #2
      }
      { \__xeCJK_check_single_save:N #1 #2#3 }
  }
\tl_new:N \l__xeCJK_check_single_cs_case_tl
```

(*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
  {
    \str_case_x:non {#4}
      { \l__xeCJK_inline_env_case_tl }
      { \use_i:nn }
      {#1} {#2}
    \scan_stop: #3 {#4}
  }
\cs_generate_variant:Nn \str_case_x:nnn { 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 { \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 { \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} { \use_ii:nn } } }
  }
\tl_new:N \l__xeCJK_inline_env_case_tl
```

(*End definition for* __xeCJK_update_inline_env_case_tl: *This function is documented on page* **??**.)

PlainEquation

```
\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 8.*)

__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。

```
\__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.
  }
```

```
\cs_if_exist:NF \XeTeXglyphbounds
  {
    \__xeCJK_error:n { XeTeX-too-old }
    \__xeCJK_after_end_preamble:n { \xeCJKsetup { PunctStyle = plain } }
  }
```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

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

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

\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

相关选项声明。

```
\keys_define:nn { xeCJK / options }
  {
    AllowBreakBetweenPuncts .choice: ,
    AllowBreakBetweenPuncts / true  .code:n =
      { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_allow_break: } ,
    AllowBreakBetweenPuncts / false .code:n =
      { \cs_set_eq:NN \__xeCJK_punct_break: \xeCJK_no_break: } ,
    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
  }
```

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

\g__xeCJK_special_punct_clist
__xeCJK_set_special_punct:nn
__xeCJK_add_special_punct:nn
__xeCJK_sub_special_punct:nn

```
\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}
      }
  }
```

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

`\__xeCJK_punct_if_right_p:N`
`\__xeCJK_punct_if_long_p:N`
`\__xeCJK_punct_if_middle_p:N`
`\__xeCJK_punct_if_mixed_width_p:N`
`\__xeCJK_punct_if_right:NTF`
`\__xeCJK_punct_if_long:NTF`
`\__xeCJK_punct_if_middle:NTF`
`\__xeCJK_punct_if_mixed_width:NTF`

```
\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:
      }
  }
```

(*End definition for* `\__xeCJK_punct_if_right:N` *and others.*)

`\__xeCJK_punct_dim_csname:nn`
`\__xeCJK_punct_dim_csname:nnn`
`\__xeCJK_punct_dim_csname:nnnn`
`\__xeCJK_use_punct_dim:nnn`
`\__xeCJK_use_punct_dim:nn`
`\__xeCJK_gset_punct_dim:nnx`
`\__xeCJK_gset_punct_dim:nnnx`
`\__xeCJK_gset_punct_dim:nnnnx`

一些用于记录的辅助函数。

```
\cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nn #1#2
  { g__\l_xeCJK_current_font_tl/\l_xeCJK_punct_style_tl/#1/#2/tl }
\cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnn
  { \__xeCJK_punct_dim_csname:nnnn { \l_xeCJK_punct_style_tl } }
\cs_new_nopar:Npn \__xeCJK_punct_dim_csname:nnnn #1#2#3#4
  { g__\l_xeCJK_current_font_tl/#1/#2/#3/#4/tl }
\cs_new_nopar:Npn \__xeCJK_use_punct_dim:nn #1#2
  { \tl_use:c { \__xeCJK_punct_dim_csname:nn {#1} {#2} } }
\cs_new_nopar:Npn \__xeCJK_use_punct_dim:nnn #1#2#3
  { \tl_use:c { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } }
\cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnx #1#2#3
  { \tl_gset:cx { \__xeCJK_punct_dim_csname:nn {#1} {#2} } {#3} }
\cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnx #1#2#3#4
  { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnn {#1} {#2} {#3} } {#4} }
\cs_new_protected_nopar:Npn \__xeCJK_gset_punct_dim:nnnnx #1#2#3#4#5
  { \tl_gset:cx { \__xeCJK_punct_dim_csname:nnnn {#1} {#2} {#3} {#4} } {#5} }
```

(*End definition for* `\__xeCJK_punct_dim_csname:nn` *and others.*)

定义标点处理模板。

```
\DeclareObjectType { xeCJK / punctuation } { \c_three }
```

```
\DeclareTemplateInterface { xeCJK / punctuation } { basic } { \c_three }
  {
    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 } ,
    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_zero_fp ,
    different-align-margin  : length  = \c_max_dim ,
    different-align-ratio   : real    = \c_zero_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_three }
  {
    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 ,
    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
    \tl_if_blank:nTF {#3}
```

```
      { \xeCJK_punct_margin_process:NN {#1} {#2} }
      { \xeCJK_punct_kerning_process:NN {#2} {#3} }
  }
```

\xeCJK_punct_margin_process:NN

```
\cs_new_protected_nopar:Npn \xeCJK_punct_margin_process:NN #1#2
  {
    \tl_set:Nx \l__xeCJK_tmpa_tl { \__xeCJK_use_punct_dim:nnn { bound } {#1} {#2} }
    \tl_set:Nx \l__xeCJK_tmpb_tl
      {
        \__xeCJK_use_punct_dim:nnn { bound }
          { \xeCJK_reverse:nnn {#1} \c__xeCJK_left_tl \c__xeCJK_right_tl } {#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 }
              {
                \xeCJK_if_blank_x: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} }
              }
          }
      }
    \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2}
      {
        \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_tmpa_tl \l__xeCJK_tmpb_tl } }
                      { \use:n }
                      {
                        \l__xeCJK_tmp_dim - ( \l__xeCJK_tmpb_tl )
                        - ( \__xeCJK_use_punct_dim:nn { dimen } {#2} )
                      }
                  }
              }
              {
                \bool_if:NTF \l__xeCJK_optimize_margin_bool
                  { \dim_min:nn { \l__xeCJK_tmpa_tl } }
                  { \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_tmpa_tl + \l__xeCJK_tmpb_tl ) / \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 } }
                        }
                    }
                }
            }
        }
    }
```

(*End definition for* `\xeCJK_punct_margin_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_tmpa_tl \scan_stop:
      }
    \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
      { + \dim_min:nn \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl }
  }
```

(*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
  {
    \__xeCJK_gset_punct_dim:nnnx { kern } {#1} {#2}
      {
        \dim_eval:n
          {
            \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_use_punct_dim:nnn { bound } \c__xeCJK_right_tl {#1} )
            - ( \__xeCJK_use_punct_dim:nnn { bound } \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 } \use_none:nn }
      {
        \fp_compare:nNnTF { \use:c { l__xeCJK_#1_align_ratio_fp } } = \c_zero_fp
          { \use:n }
          { \fp_use:c { l__xeCJK_#1_align_ratio_fp } \use_ii:nn }
      }
      {
        \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 }
      }
      { \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_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_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim }
            \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl  {#2} { \c_zero_dim }
            \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
          }
          {
            { \xeCJK_select_font: \xeCJK_calc_punct_dimen:f {#2} }
            \__xeCJK_punct_if_long:NTF {#2}
              { \__xeCJK_gset_punct_dim:nnnx { glue } {#1} {#2} { \c_zero_dim } }
              {
                \UseInstance { xeCJK / punctuation }
                  { \l_xeCJK_punct_style_tl } {#1} {#2} { }
              }
          }
      }
  }
```

(*End definition for* \xeCJK_get_punct_bounds:NN)

\xeCJK_calc_punct_dimen:N

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

```
\cs_new_protected_nopar:Npn \xeCJK_calc_punct_dimen:N #1
  {
    \__xeCJK_gset_punct_dim:nnnx { bound } \c__xeCJK_left_tl {#1}
      { \xeCJK_glyph_bounds:NN \c_one {#1} }
    \__xeCJK_gset_punct_dim:nnnx { 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_gset_punct_dim:nnx { width } {#1}
      { \dim_use:N \etex_fontcharwd:D \tex_font:D \xeCJK_token_value_charcode:N #1 }
    \__xeCJK_gset_punct_dim:nnx { dimen } {#1}
      { \dim_eval:n { \__xeCJK_use_punct_dim:nn { width } {#1} - \l__xeCJK_tmp_dim } }
    \__xeCJK_punct_if_long:NT {#1}
      {
        \seq_map_inline:Nn \g__xeCJK_punct_style_seq
          {
            \__xeCJK_gset_punct_dim:nnnnx {##1} { kern } {#1} {#1}
              {
                \bool_if:nTF
                  {
                    \str_if_eq_p:nn {#1} { ^^^^2025 } ||
                    \str_if_eq_p:nn {#1} { ^^^^2026 }
                  }
                  { \c_zero_dim }
                  { - \dim_use:N \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)

\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_gset_punct_dim:nnnx { kern } {#1} {#2} { \c_zero_dim } }
          {
            \UseInstance { xeCJK / punctuation }
              { \l_xeCJK_punct_style_tl } { } {#1} {#2}
          }
      }
  }
\cs_generate_variant:Nn \xeCJK_get_punct_kerning:NN { o }
```

(*End definition for* \xeCJK_get_punct_kerning: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_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:Nn \g__xeCJK_punct_style_seq {#1} }
    \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 9.*)

`\xeCJKEditPunctStyle` 对已有的标点处理风格进行修改。

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

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

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_fallback_loop:Nn {#1} { \l_xeCJK_family_tl/FallBack } } }
  }
```

(*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。

```
\cs_new_protected_nopar:Npn \xeCJK_fallback_loop:Nn #1#2
  {
    \xeCJK_family_if_exist:xTF {#2}
      {
        \xeCJK_select_font:x {#2}
        \xeCJK_glyph_if_exist:NTF {#1}
          { \__xeCJK_fallback_save_CJKsymbol:N {#1} }
          { \xeCJK_fallback_loop:Nn {#1} { #2/FallBack } }
      }
      {
        \xeCJK_family_if_exist:xT { \l_xeCJK_family_tl/FallBack }
          { \__xeCJK_warning:nxx { fallback } {#1} {#2} }
        \__xeCJK_fallback_save_CJKsymbol:N {#1}
      }
  }
\__xeCJK_msg_new:nn { fallback }
  {
    CJKfamily~'#2'~
    ( \prop_get:No \g__xeCJK_family_font_name_prop {#2} )~
    does~not~contain~glyph~'#1'~(U+\int_to_hexadecimal:n {`#1}).\\
  }
\bool_new:N \l__xeCJK_fallback_first_bool
\cs_generate_variant:Nn \prop_get:Nn { No }
```

(*End definition for* \xeCJK_fallback_loop:Nn)

\setCJKfallbackfamilyfont

```
\NewDocumentCommand \setCJKfallbackfamilyfont { m O{} m }
  {
    \clist_set:Nx \l__xeCJK_tmpa_clist {#3}
    \int_compare:nNnTF { \clist_count:N \l__xeCJK_tmpa_clist } > \c_one
      { \xeCJK_set_family_fallback:xxN {#1} {#2} \l__xeCJK_tmpa_clist }
      { \xeCJK_set_family:xxx { #1/FallBack } {#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:cn
          { l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist } {#2}
        \xeCJK_set_family:Vvv \l__xeCJK_fallback_family_tl
          { l__xeCJK_ \l__xeCJK_fallback_family_tl _font_options_clist }
          { l__xeCJK_ \l__xeCJK_fallback_family_tl _font_name_tl }
      }
    \group_end:
  }
\cs_generate_variant:Nn \xeCJK_set_family_fallback:nnN { xx }
```

(*End definition for* `\xeCJK_set_family_fallback:nnN`)

5.12 CJK 字体族声明方式

```
\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}
              {
                \bool_set_false:c { l__xeCJK_copy_#1_bool }
                \bool_set_false:c { l__xeCJK_add_#1_bool }
                \tl_put_right:Nn \l__xeCJK_family_name_tl { /#1 }
              }
```

```
            {
              \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
              \str_if_eq:nnTF {##1} { * }
                {
                  \bool_set_true:c  { l__xeCJK_copy_#1_bool }
                  \bool_set_false:c { l__xeCJK_add_#1_bool }
                }
                {
                  \bool_set_false:c { l__xeCJK_copy_#1_bool }
                  \bool_set_true:c  { l__xeCJK_add_#1_bool }
                  \__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:wn

```
\cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:nn #1#2
  {
    \tl_set:Nx \l__xeCJK_tmpa_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
    \clist_clear:N \l__xeCJK_sub_font_options_clist
    \tl_if_head_eq_meaning:VNTF \l__xeCJK_tmpa_tl [ % ]
      {
        \exp_after:wN \__xeCJK_get_sub_features:wn \l__xeCJK_tmpa_tl \c_empty_tl
        \tl_if_blank:VT \l__xeCJK_sub_font_name_tl
          {
            \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl
            \clist_clear:N \l__xeCJK_sub_font_options_clist
          }
      }
      { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmpa_tl }
    \tl_if_blank:VTF \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 }
    \clist_set_eq:cN { l__xeCJK_#1_font_options_clist } \l__xeCJK_sub_font_options_clist
    \tl_set_eq:cN { l__xeCJK_#1_font_name_tl } \l__xeCJK_sub_font_name_tl
  }
\cs_new_protected_nopar:Npn \__xeCJK_get_sub_features:wn [#1] #2
  {
    \clist_set:Nn \l__xeCJK_sub_font_options_clist {#1}
    \str_if_eq:nnTF {#2} { \c_empty_tl }
      { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \c_empty_tl }
      { \tl_set:Nn \l__xeCJK_sub_font_name_tl {#2} }
  }
\cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { V }
\cs_generate_variant:Nn \tl_if_head_eq_meaning:nNTF { V }
\cs_generate_variant:Nn \tl_replace_all:Nnn { NnV }
```

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

FallBack

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

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

\g__xeCJK_features_id_prop
BoldFont
ItalicFont
BoldItalicFont
SlantedFont
BoldSlantedFont
BoldFeatures
ItalicFeatures
BoldItalicFeatures

调用字体的属性声明，同 fontspec 宏包。

```
\prop_new:N \g__xeCJK_features_id_prop
\prop_put:Nnn \g__xeCJK_features_id_prop { bf   } { Bold       }
\prop_put:Nnn \g__xeCJK_features_id_prop { it   } { Italic     }
\prop_put:Nnn \g__xeCJK_features_id_prop { bfit } { BoldItalic }
\prop_put:Nnn \g__xeCJK_features_id_prop { sl   } { Slanted    }
```

```
\prop_put:Nnn \g__xeCJK_features_id_prop { bfsl } { BoldSlanted }
\prop_map_inline:Nn \g__xeCJK_features_id_prop
  {
    \keys_define:nn { xeCJK / features }
      {
        #2Font        .tl_set:c = { l__xeCJK_font_name_#1_tl } ,
        #2Features .clist_set:c = { l__xeCJK_font_feat_#1_clist }
      }
  }
```

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

AutoFakeBold
AutoFakeSlant

```
\keys_define:nn { xeCJK / features }
  {
    AutoFakeBold  .choice: ,
    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  = { \g__xeCJK_embolden_factor_fp } ,
    AutoFakeSlant  .choice: ,
    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  = { \g__xeCJK_slant_factor_fp }
  }
```

(*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_add_font_prop
    \prop_map_inline:Nn \g__xeCJK_features_id_prop
      {
        \tl_clear:c { l__xeCJK_font_name_##1_tl }
        \clist_clear:c { l__xeCJK_font_feat_##1_clist }
      }
    \clist_clear:N \l__xeCJK_fontspec_options_clist
    \seq_clear:N \l__xeCJK_sub_key_seq
    \seq_map_inline:Nn \g__xeCJK_sub_key_seq
      {
        \bool_set_false:c { l__xeCJK_copy_##1_bool }
        \bool_set_false:c { l__xeCJK_add_##1_bool  }
      }
    \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
```

(*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_features_clist
      \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
    \__xeCJK_remove_duplicate_keys:N \l__xeCJK_font_features_clist
    \keys_set_known:nVN { xeCJK / features }
      \l__xeCJK_font_features_clist \l__xeCJK_fontspec_options_clist
    \__xeCJK_parse_features:
    \__xeCJK_pass_features:
    \__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:
  }
\cs_generate_variant:Nn \xeCJK_set_family:nnn { Vnn , xxx , Vvv }
```

(*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_tmp_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_tmp_prop
      {
        \tl_if_blank:nTF {##2}
          { \clist_put_right:Nn #1 {##1} }
          { \clist_put_right:Nn #1 { ##1 = {##2} } }
      }
  }
\cs_generate_variant:Nn \keyval_parse:NNn { NNV }
\cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:n #1
  { \prop_put:Nnn \l__xeCJK_tmp_prop {#1} { } }
\cs_new_protected_nopar:Npn \__xeCJK_prop_put_aux:nn #1#2
  { \prop_put:Nnn \l__xeCJK_tmp_prop {#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:
        \cs_set_eq:NN
        \exp_not:n { \__xeCJK_update_family:n \use_none:n }
        \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:
      }
  }
```

(*End definition for* __xeCJK_gset_family_cs:x)

__xeCJK_check_family:n

```
\cs_new_protected_nopar:Npn \__xeCJK_check_family:n #1
  {
```

```
    \prop_get:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmpa_tl
      {
        \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmpa_tl
          { \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} } }
        \__xeCJK_warning:nxx { CJKfamily-redef } {#1} { \l__xeCJK_tmpa_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_add_font:nn`
`\__xeCJK_add_font_if_new:nn`

```
\cs_new_protected_nopar:Npn \__xeCJK_add_font:nn #1#2
  { \prop_put:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
\cs_new_protected_nopar:Npn \__xeCJK_add_font_if_new:nn #1#2
  { \prop_put_if_new:Nnn \l__xeCJK_add_font_prop { #1 Font } {#2} }
\cs_generate_variant:Nn \__xeCJK_add_font:nn        { nx , nV , nv }
\cs_generate_variant:Nn \__xeCJK_add_font_if_new:nn { nx , nV , nv }
```

(*End definition for* `\__xeCJK_add_font:nn` *and* `\__xeCJK_add_font_if_new:nn`)

`\__xeCJK_add_fake_bold:n`
`\__xeCJK_add_fake_slant:n`

```
\cs_new_protected_nopar:Npn \__xeCJK_add_fake_bold:n #1
  {
    \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
      { FakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
  }
\cs_new_protected_nopar:Npn \__xeCJK_add_fake_slant:n #1
  {
    \clist_put_left:cx { l__xeCJK_font_feat_#1_clist }
      { FakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
  }
```

(*End definition for* `\__xeCJK_add_fake_bold:n` *and* `\__xeCJK_add_fake_slant:n`)

`\__xeCJK_parse_features:`
`\__xeCJK_parse_features:nn`

```
\cs_new_protected_nopar:Npn \__xeCJK_parse_features:
  { \prop_map_function:NN \g__xeCJK_features_id_prop \__xeCJK_parse_features:nn }
\cs_new_protected_nopar:Npn \__xeCJK_parse_features:nn #1#2
  {
    \__xeCJK_if_font_select:nTF {#1}
      {
        \__xeCJK_add_font:nv {#2} { l__xeCJK_font_name_#1_tl }
        \__xeCJK_if_it_or_sl:nTF {#1}
          {
            \__xeCJK_if_font_select:nF { bf #1 }
              { \__xeCJK_add_font_if_new:nv {#2} { l__xeCJK_font_name_#1_tl }  }
          }
          {
            \str_if_eq:nnT {#1} { bf }
              {
                \__xeCJK_if_font_select:nF { bfit }
                  { \__xeCJK_add_font:nV { BoldItalic } \l__xeCJK_font_name_bf_tl }
                \__xeCJK_if_font_select:nF { bfsl }
                  { \__xeCJK_add_font:nV { BoldSlanted } \l__xeCJK_font_name_bf_tl }
              }
          }
      }
      { \__xeCJK_set_fake_font:nn {#1} {#2} }
  }
```

(*End definition for* `\__xeCJK_parse_features:` *and* `\__xeCJK_parse_features:nn`)

__xeCJK_if_font_select_p:n
__xeCJK_if_it_or_sl_p:n
__xeCJK_if_font_select:nTF
__xeCJK_if_it_or_sl:nTF

```
\prg_new_conditional:Npnn \__xeCJK_if_font_select:n #1 { p , T , F , TF }
  {
    \tl_if_blank:vTF { l__xeCJK_font_name_#1_tl }
      { \prg_return_false: } { \prg_return_true: }
  }
\prg_new_conditional:Npnn \__xeCJK_if_it_or_sl:n #1 { p , T , F , TF }
  {
    \if_predicate:w
      \bool_if_p:n { \str_if_eq_p:nn { it } {#1} || \str_if_eq_p:nn { sl } {#1} }
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\cs_generate_variant:Nn \tl_if_blank:nTF { v }
```

(*End definition for* __xeCJK_if_font_select:n *and* __xeCJK_if_it_or_sl:n)

__xeCJK_set_fake_font:nn

```
\cs_new_protected_nopar:Npn \__xeCJK_set_fake_font:nn #1#2
  {
    \str_if_eq:nnTF {#1} { bf }
      { \bool_if:NT \l__xeCJK_auto_fake_bold_bool { \__xeCJK_add_fake_bold:n {#1} } }
      {
        \bool_if:NTF \l__xeCJK_auto_fake_slant_bool
          {
            \bool_if:nT
              {    \__xeCJK_if_it_or_sl_p:n {#1}
                || ( \str_if_eq_p:nn {#1} { bfit }
                     && ! ( \__xeCJK_if_font_select_p:n { it } ) )
                || ( \str_if_eq_p:nn {#1} { bfsl }
                     && ! ( \__xeCJK_if_font_select_p:n { sl } ) )
              }
              { \__xeCJK_add_fake_slant:n {#1} }
          }
          { \__xeCJK_if_it_or_sl:nT {#1} { \__xeCJK_map_it_sl:nn {#1} {#2} } }
        \bool_if:nT
          {      \l__xeCJK_auto_fake_bold_bool
            && ! ( \__xeCJK_if_it_or_sl_p:n {#1} )
            && ! ( \__xeCJK_if_font_select_p:n { bf } )
          }
          { \__xeCJK_add_fake_bold:n {#1} }
      }
    \__xeCJK_add_font_if_new:nn {#2} { * }
  }
```

(*End definition for* __xeCJK_set_fake_font:nn)

__xeCJK_map_it_sl:n

```
\cs_new_protected_nopar:Npn \__xeCJK_map_it_sl:nn #1#2
  {
    \__xeCJK_if_map_font_select:nT {#1}
      {
        \__xeCJK_add_font:nx {#2} { \__xeCJK_get_map_font:n {#1} }
        \__xeCJK_if_font_select:nF { bf #1 }
          { \__xeCJK_add_font_if_new:nx { Bold #2 } { \__xeCJK_get_map_font:n {#1} } }
      }
  }
```

(*End definition for* __xeCJK_map_it_sl:n)

__xeCJK_get_map_font:n

```
\cs_new_nopar:Npn \__xeCJK_get_map_font:n #1
  { \exp_not:v { l__xeCJK_font_name_ \xeCJK_reverse:nnn {#1} { it } { sl } _tl } }
```

(*End definition for* __xeCJK_get_map_font:n)

__xeCJK_if_map_font_select_p:n
__xeCJK_if_map_font_select:nTF

```
\prg_new_conditional:Npnn \__xeCJK_if_map_font_select:n #1 { p , T , F , TF }
  {
    \xeCJK_if_blank_x:nTF { \__xeCJK_get_map_font:n {#1} }
      { \prg_return_false: } { \prg_return_true: }
  }
```

(*End definition for* __xeCJK_if_map_font_select:n)

__xeCJK_pass_features:

```
\cs_new_protected_nopar:Npn \__xeCJK_pass_features:
  {
    \prop_map_inline:Nn \g__xeCJK_features_id_prop
      {
        \clist_if_empty:cF { l__xeCJK_font_feat_##1_clist }
          {
            \clist_put_right:Nx \l__xeCJK_fontspec_options_clist
              { ##2Features = { \exp_not:v { l__xeCJK_font_feat_##1_clist } } }
          }
      }
    \prop_map_inline:Nn \l__xeCJK_add_font_prop
      { \clist_put_right:Nn \l__xeCJK_fontspec_options_clist { ##1 = {##2} } }
  }
```

(*End definition for* __xeCJK_pass_features:)

\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_prop)

__xeCJK_save_family_info:

```
\cs_new_protected_nopar:Npn \__xeCJK_save_family_info:
  {
    \group_begin:
    \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
      { \clist_remove_all:Nn \l__xeCJK_font_options_clist {##1} }
    \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
    \group_end:
  }
```

(*End definition for* __xeCJK_save_family_info:)

__xeCJK_set_sub_block_family:

```
\cs_new_protected_nopar:Npn \__xeCJK_set_sub_block_family:
  {
    \seq_map_inline:Nn \l__xeCJK_sub_key_seq
      {
        \tl_set:Nx \l__xeCJK_sub_family_tl { \l__xeCJK_family_name_tl/##1 }
        \bool_if:cT { l__xeCJK_copy_##1_bool } { \__xeCJK_copy_sub_family:n {##1} }
        \bool_if:cT { l__xeCJK_add_##1_bool }
          {
            \xeCJK_set_family:Vvv \l__xeCJK_sub_family_tl
              { l__xeCJK_##1_font_options_clist } { l__xeCJK_##1_font_name_tl }
          }
      }
  }
\cs_new_protected_nopar:Npn \__xeCJK_copy_sub_family:n #1
  {
    \__xeCJK_check_family:V \l__xeCJK_sub_family_tl
    \prop_get:NVNT \g__xeCJK_family_font_name_prop
```

```
      \l__xeCJK_family_name_tl \l__xeCJK_tmpa_tl
      {
        \prop_gput:NVV \g__xeCJK_family_font_name_prop
          \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_tl
      }
    \prop_get:NVNT \g__xeCJK_family_font_options_prop
      \l__xeCJK_family_name_tl \l__xeCJK_tmpa_clist
      {
        \clist_remove_all:Nn \l__xeCJK_tmpa_clist { #1 = * }
        \prop_gput:NVV \g__xeCJK_family_font_options_prop
          \l__xeCJK_sub_family_tl \l__xeCJK_tmpa_clist
      }
    \cs_gset_protected_nopar:cpx { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_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_tmpa_tl
              {
                \__xeCJK_gset_family_nfss_cs:xx
                  { \l__xeCJK_sub_family_tl } { \exp_not:N \l__xeCJK_tmpa_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_tmpa_tl
              { \prop_gput:NnV ##1 {#1} \l__xeCJK_tmpa_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 }
      }
  }
\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} }
      {
        \tl_set:Nx \exp_not:N \f@encoding { \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
  }
```

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

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

```
\tl_new:N \l_xeCJK_family_tl
\tl_set:Nn \l_xeCJK_family_tl { \CJKfamilydefault }
```

(*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`
`\__xeCJK_family_unknown_warning:n`

```
\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} }
  }
\cs_new_protected_nopar:Npn \__xeCJK_family_unknown_warning:n #1
  {
    \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~
    \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} }
    [...]{...}~to~define~it.
  }
\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_if_exist_use:x` *and* `\__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} }
\cs_new_eq:NN \setCJKromanfont \setCJKmainfont
\NewDocumentCommand \setCJKsansfont { O{} m }
  { \xeCJK_set_family:xxx { \CJKsfdefault } {#1} {#2} }
\NewDocumentCommand \setCJKmonofont { O{} m }
  { \xeCJK_set_family:xxx { \CJKttdefault } {#1} {#2} }
```

(*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_tmpa_tl { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
    \cs_new_protected_nopar:Npx #2 { \CJKfamily { \l__xeCJK_tmpa_tl } }
    \xeCJK_set_family:xxx { \l__xeCJK_tmpa_tl } {#3} {#4}
  }
\NewDocumentCommand \CJKfontspec { O{} m }
  {
    \xeCJK_fontspec:xx {#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
  {
    \tl_set:Nx \l__xeCJK_tmpa_tl { \tl_to_str:n { CJKfontspec/#1/#2/id } }
    \tl_if_exist:cF { \l__xeCJK_tmpa_tl }
      {
        \tl_set:Nx \l__xeCJK_tmpb_tl
          { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + \c_one } ) }
        \tl_gset_eq:cN { \l__xeCJK_tmpa_tl } \l__xeCJK_tmpb_tl
        \xeCJK_set_family:Vnn \l__xeCJK_tmpb_tl {#1} {#2}
      }
    \exp_args:Nc \CJKfamily { \l__xeCJK_tmpa_tl }
  }
\cs_generate_variant:Nn \xeCJK_fontspec:nn { VV , xx }
```

(*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 { m }
  {
    \xeCJK_add_font_features:x {#1}
    \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:n

```
\cs_new_protected_nopar:Npn \xeCJK_add_font_features:n #1
  {
    \prop_get:NVNTF \g__xeCJK_family_font_name_prop \l_xeCJK_family_tl \l__xeCJK_tmpa_tl
      {
        \clist_set:Nn \l__xeCJK_tmpa_clist {#1}
        \seq_map_function:NN
          \g__xeCJK_CJK_sub_class_seq \__xeCJK_parse_sub_class_features:n
        \prop_get:NVNT \g__xeCJK_family_font_options_prop
          \l_xeCJK_family_tl \l__xeCJK_tmpb_clist
          {
            \clist_concat:NNN
              \l__xeCJK_tmpa_clist \l__xeCJK_tmpb_clist \l__xeCJK_tmpa_clist
          }
        \xeCJK_fontspec:VV \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_tl
      }
      { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
  }
\cs_generate_variant:Nn \xeCJK_add_font_features:n { x }
\__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:n)

__xeCJK_parse_sub_class_features:n

```
\cs_new_protected_nopar:Npn \__xeCJK_parse_sub_class_features:n #1
  {
    \clist_if_in:NnT \l__xeCJK_tmpa_clist {#1}
      {
        \clist_remove_all:Nn \l__xeCJK_tmpa_clist {#1}
        \prop_get:NoNF \g__xeCJK_family_font_name_prop
```

```
          { \l_xeCJK_family_tl/#1 } \l__xeCJK_tmpb_tl
          {
            \prop_get:NxNF \g__xeCJK_family_font_name_prop
              { \CJKfamilydefault/#1 } \l__xeCJK_tmpb_tl
              { \tl_set_eq:NN \l__xeCJK_tmpb_tl \l__xeCJK_tmpa_tl }
          }
        \clist_set:Nx \l__xeCJK_tmpa_clist
         { #1 = { [ \exp_not:V \l__xeCJK_tmpa_clist ] { \exp_not:V \l__xeCJK_tmpb_tl } } }
        \clist_map_break:
      }
  }
\cs_generate_variant:Nn \prop_get:NnNF { Nx }
```

(*End definition for* `\__xeCJK_parse_sub_class_features:n`)

在导言区结束的时候，若没有声明 CJK 字体，则给出一个警告。并在此时根据西文字体的情况更新 `\CJKfamilydefault` 和设置数学字体。

```
\__xeCJK_at_end_preamble:n
  {
    \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
      { \__xeCJK_warning:n { no-CJKfamily } }
      {
        \tl_gset:Nx \CJKfamilydefault
          {
            \str_case_x:nnn { \familydefault }
              {
                { \rmdefault } { \CJKrmdefault }
                { \sfdefault } { \CJKsfdefault }
                { \ttdefault } { \CJKttdefault }
              }
              { \CJKfamilydefault }
          }
        \CJKfamily { \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\\\\
    \token_to_str:N \xeCJKmainfont[...]{...}\\\\
    in~the~preamble~to~declare~the~main~CJKfamily.\\
  }
```

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* 6.)

`\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_tmpa_tl { \c__xeCJK_math_tl } }
      {
        \xeCJK_family_if_exist:xTF { \CJKfamilydefault }
```

```
          { \tl_set:Nx \l__xeCJK_tmpa_tl { \CJKfamilydefault } }
          { \use_none:nnnnn }
      }
    \prop_get:NVNT \g__xeCJK_family_name_prop \l__xeCJK_tmpa_tl \l__xeCJK_tmpb_tl
      {
        \tl_const:Nx \c__xeCJK_math_family_tl { \l__xeCJK_tmpb_tl }
        \DeclareSymbolFont \c__xeCJK_math_tl
          \c__xeCJK_encoding_tl \c__xeCJK_math_family_tl \mddefault \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_tmpa_clist \c__xeCJK_CJK_chars_clist \c__xeCJK_FullLeft_chars_clist
        \clist_concat:NNN
          \l__xeCJK_tmpa_clist \l__xeCJK_tmpa_clist \c__xeCJK_FullRight_chars_clist
        \clist_map_inline:Nn \l__xeCJK_tmpa_clist
          {
            \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn {##1}
              { \c_seven } { \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_tmpa_int \l__xeCJK_tmpb_int
    \xeCJK_int_until_do:nn { \l__xeCJK_tmpa_int > \l__xeCJK_tmpb_int }
      {
        \tex_global:D \XeTeXmathcode \l__xeCJK_tmpa_int = #3 ~ #4 \l__xeCJK_tmpa_int
        \int_incr:N \l__xeCJK_tmpa_int
      }
  }
```

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

5.15 抄录环境中的间距调整

`\xeCJKVerbAddon`

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

```
\NewDocumentCommand \xeCJKVerbAddon { }
  {
    \int_compare:nNnF \etex_currentgrouplevel:D = \c_zero
      {
        \__xeCJK_set_verb_exspace:
        \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 }
            \xeCJKsetup { xCJKecglue = false }
            \cs_set_protected_nopar:Npn \CJKglue
              { \skip_horizontal:N \g__xeCJK_verb_exspace_skip }
            \cs_set_protected_nopar:Npn \CJKecglue
              { \skip_horizontal:n { 0.5 \g__xeCJK_verb_exspace_skip } }
            \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
            \cs_set_eq:NN \xeCJK_ignore_spaces:w \CJKecglue
            \skip_set:Nn \tex_rightskip:D { \c_zero_dim plus 1 fil }
```

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

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

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

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

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

```
\cs_new_protected_nopar:Npn \__xeCJK_set_verb_exspace:
  {
    \tl_if_exist:cF { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size }
      {
        \group_begin:
        \xeCJK_select_font:
        \use:x
          {
            \group_end:
            \tl_gset:cx
            \exp_not:n { { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
              {
                \exp_not:N \dim_max:nn { \c_zero_dim }
                  {
                    \c_two \tex_fontdimen:D \c_two \tex_font:D -
                    \dim_use:N \etex_fontcharwd:D \tex_font:D "4E00
                  }
              }
          }
      }
    \skip_gset:Nn \g__xeCJK_verb_exspace_skip
      { \use:c { xeCJK/verb/\l_xeCJK_family_tl/\curr@fontshape/\f@size } }
  }
\skip_new:N \g__xeCJK_verb_exspace_skip
```

(*End definition for* __xeCJK_set_verb_exspace:)

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

```
\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: }
      \textvisiblespace
  } }
\AtEndOfPackage
  {
    \tl_put_left:Nn \fontspec_visible_space: { \scan_stop: }
    \cs_gset_eq:NN \fontspec_visible_space_fallback: \xeCJK_visible_space_fallback:
  }
```

(*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
  {
    \tl_set:Nx \f@encoding { \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.*)

CJKnumber indentfirst 是否启用 CJKnumb 宏包和首行是否缩进。

```
\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* `CJKnumber` *and* `indentfirst` *These functions are 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  ,
    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 ^^^^2500 ^^^^00b7 ^^^^30fb ^^^^ff65 } ,
    AllowBreakBetweenPuncts = false
  }
\defaultCJKfontfeatures { Script = CJK }
```

执行宏包选项，并载入 fontspec 宏包。

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

\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_if_exist:NF \CJKfamilydefault { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
\tl_new:c { \__xeCJK_family_csname:n { \CJKfamilydefault } }
```

(*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_msg_new:nn { conflict-package }
  {
    The~`#1'~package~is~in~conflict~with~xeCJK. \\
    Please~do~not~use~it~or~load~it~after~xeCJK.
  }
\xeCJK_if_package_loaded:nTF { realscripts }
  {
    \@ifpackagelater { realscripts } { 2010/10/10 } { }
      { \__xeCJK_error:nx { conflict-package } { xltxtra } }
  }
  {
    \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:n
      {
        \str_case_x:nnn {#1}
          {
            { \rmdefault }     { \CJKfamily { \CJKrmdefault } }
            { \sfdefault }     { \CJKfamily { \CJKsfdefault } }
```

```
            { \ttdefault }     { \CJKfamily { \CJKttdefault } }
            { \familydefault } { \CJKfamily { \CJKfamilydefault } }
          }
          { }
      }
  }
\cs_new_eq:NN \__xeCJK_update_family:n \use:n
```

(*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 }
          {
            \tex_unkern:D \tex_unkern:D \@@italiccorr
            { \xeCJK_make_node:n { default } }
          }
          {
            \xeCJK_if_last_node:nTF { CJK }
              {
                \tex_unkern:D \tex_unkern:D \@@italiccorr
                { \xeCJK_make_node:n { CJK } } \use:n
              }
              {
                \xeCJK_if_last_node:nTF { CJK-space }
                  {
                    \tex_unkern:D \tex_unkern:D \@@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)

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

```
\__xeCJK_after_preamble:n
  {
    \tl_map_inline:nn
      {
        \textellipsis  \textemdash     \textperiodcentered \textcentereddot
        \textquoteleft \textquoteright \textquotedblleft   \textquotedblright
      }
      { \cs_gset_nopar:Npx #1 { { \makexeCJKinactive \exp_not:o {#1} } } }
    \tl_put_left:Nn \tipaencoding { \makexeCJKinactive }
    \cs_new_eq:NN \__xeCJK_aux_r:n \r
    \cs_set_nopar:Npn \r #1 { { \makexeCJKinactive \__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 }
      {
        \tl_gput_right:Nn \pdfstringdefPreHook
          {
            \__xeCJK_gobble_CJKfamily:
            \xeCJK_cs_clear:N \makexeCJKinactive
          }
      }
  }
```

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 }
  }
```

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

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

\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。

```
\cs_if_eq:NNTF \ifCTEX@fntef \tex_iftrue:D
  { \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 } }
              }
          }
      }
  }
```

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。

```
\bool_if:NT \g__xeCJK_number_bool
  {
    \tl_set:Nn \CJK@UnicodeEnc { UTF8 }
    \cs_set_protected:Npn \CJKaddEncHook #1#2
```

```
      { \cs_set:cpn { __xeCJK_enc_#1_hook: } {#2} }
    \cs_set_protected_nopar:Npn \Unicode #1#2
      { \tex_char:D \etex_numexpr:D (#1) * \c_two_hundred_fifty_six + (#2) \scan_stop: }
    \RequirePackage { CJKnumb }
    \cs_if_exist_use:c { __xeCJK_enc_ \CJK@UnicodeEnc _hook: }
    \tl_set:Nn \CJK@tenthousand    { \Unicode { 78 } { 7 } }
    \tl_set:Nn \CJK@hundredmillion { \Unicode { 78 } { 191 } }
  }
```

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:
\@ifpackageloaded { xeCJK }
  { }
  { \RequirePackage { xeCJK } }
\RequirePackage { ulem }
\RequirePackage { CJKfntef }
\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 }
        \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:
  {
    \cs_set_eq:NN \__xeCJK_punct_hskip:n    \__xeCJK_ulem_hskip:n
    \cs_set_eq:NN
      \__xeCJK_Default_and_FullLeft_glue:N  \__xeCJK_ulem_Default_and_FullLeft_glue:N
    \cs_set_eq:NN
      \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N
    \cs_set_eq:NN
      \__xeCJK_CJK_and_FullLeft_glue:N      \__xeCJK_ulem_CJK_and_FullLeft_glue:N
    \cs_set_eq:NN
      \__xeCJK_CJK_and_FullRight_glue:N     \__xeCJK_ulem_CJK_and_FullRight_glue:N
    \cs_set_eq:NN
      \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
    \cs_set_eq:NN \xeCJK_FullLeft_and_Default:  \__xeCJK_ulem_FullLeft_and_Default:
    \cs_set_eq:NN \xeCJK_FullLeft_and_CJK:      \__xeCJK_ulem_FullLeft_and_CJK:
    \cs_set_eq:NN \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
    \cs_set_eq:NN \xeCJK_FullRight_and_CJK:     \__xeCJK_ulem_FullRight_and_CJK:
    \cs_set_eq:NN \xeCJK_CJK_and_CJK:N          \__xeCJK_ulem_CJK_and_CJK:N
    \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} }
              }
          }
      }
  }
```

(*End definition for* __xeCJK_ulem_initial:)

__xeCJK_ulem_CJK_and_CJK:N

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_CJK_and_CJK:N
  {
    \xeCJK_class_group_end:
    \UL@stop \__xeCJK_ulem_ccglue: \UL@start
    \__xeCJK_ulem_class_group_begin:
    \CJKsymbol
  }
```

(*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_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
  }
```

(*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
  {
    \UL@stop
    \__xeCJK_ulem_skip_punct_begin:
    \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
    \UL@start
  }
```

(*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
  {
    \UL@stop
    \__xeCJK_ulem_skip_punct_begin:
    \int_compare:nNnF \etex_lastnodetype:D = \c_one
      { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1} }
    \UL@start
  }
```

(*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_class_group_end:
    \UL@stop
    \__xeCJK_ulem_skip_punct_begin:
    \__xeCJK_ulem_ccglue:
    \__xeCJK_punct_glue:NN \c__xeCJK_left_tl {#1}
```

```
    \UL@start
    \__xeCJK_ulem_class_group_begin:
  }
```

(*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
  {
    \UL@stop
    \__xeCJK_ulem_skip_punct_begin:
    \__xeCJK_punct_if_long:NTF {#1}
      { \__xeCJK_ulem_ccglue: }
      {
        \__xeCJK_punct_if_middle:NT {#1}
          {
            \xeCJK_no_break:
            \__xeCJK_punct_glue:NN \c__xeCJK_right_tl {#1}
          }
        \xeCJK_no_break:
      }
    \UL@start
  }
```

(*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_class_group_end:
    \__xeCJK_ulem_Default_and_FullRight_glue:N {#1}
    \__xeCJK_ulem_class_group_begin:
  }
```

(*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_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_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
  }
```

(*End definition for* __xeCJK_ulem_FullLeft_and_Default:)

__xeCJK_ulem_FullLeft_and_CJK:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullLeft_and_CJK:
  {
    \__xeCJK_ulem_FullLeft_and_Default:
    \__xeCJK_ulem_class_group_begin:
  }
```

(*End definition for* __xeCJK_ulem_FullLeft_and_CJK:)

__xeCJK_ulem_FullRight_and_Default:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_Default:
  {
    \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
```

```
    \xeCJK_class_group_end:
    \UL@stop
    \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \__xeCJK_ulem_skip_punct_end:
    \UL@start
  }
```

(*End definition for* __xeCJK_ulem_FullRight_and_Default:)

__xeCJK_ulem_FullRight_and_CJK:

```
\cs_new_protected_nopar:Npn \__xeCJK_ulem_FullRight_and_CJK:
  {
    \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
    \xeCJK_class_group_end:
    \UL@stop
    \__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:
  }
```

(*End definition for* __xeCJK_ulem_FullRight_and_CJK:)

__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
  {
    \cs_if_eq:NNTF \  \LA@space
      { \skip_horizontal:n {#1} }
      { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
  }
\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 }
    \CJK@preUnderdot
    \__xeCJK_make_under_symbol:n { \CJK@underdotSkip }
    \cs_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
    \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
    \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_dot_CJKsymbol:N
    \CJK@postUnderdot
    \bool_if:NT \l__xeCJK_ulem_hook_used_bool { \UL@start }
    \tex_ignorespaces:D
  }
\box_new:N \g__xeCJK_under_symbol_box
```

(*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_gset_eq:NN \__xeCJK_save_under_dot_CJKsymbol:N \CJKsymbol
    \cs_set_eq:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
    #3
    \group_end:
    \tex_ignorespaces:D
  }
```

(*End definition for* \CJKunderanysymbol)

__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
  {
    \__xeCJK_save_under_dot_CJKsymbol:N {#1}
    \hbox_overlap_left:n { \box_use:N \g__xeCJK_under_symbol_box }
    { \xeCJK_make_node:n { CJK } }
    \xeCJK_ignore_spaces:w
  }
```

(*End definition for* __xeCJK_under_CJKsymbol:N)

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

```
\RenewDocumentEnvironment { CJKfilltwosides } { m }
  {
    \dim_set:Nn \l__xeCJK_fill_width_dim {#1}
    \cs_set_eq:NN \CJKglue \tex_hfil:D
    \cs_set_eq:NN \\ \__xeCJK_fill_newline:
    \mode_if_vertical:T { \tex_indent:D }
    \tex_vbox:D \c_group_begin_token
    \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
    \tex_ignorespaces:D
  }
  {
    \tex_unskip:D
    \c_group_end_token
    \c_group_end_token
  }
\dim_new:N \l__xeCJK_fill_width_dim
```

(*End definition for* CJKfilltwosides)

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

```
\cs_new_protected_nopar:Npn \__xeCJK_fill_newline:
  {
    \tex_unskip:D
    \c_group_end_token
```

```
    \tex_hbox:D to \l__xeCJK_fill_width_dim \c_group_begin_token
    \tex_ignorespaces:D
  }
```

(*End definition for* `\__xeCJK_fill_newline:`)

⟨/fntef⟩

5.20 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。 40

删除多余的 default-itcorr 结点。 25

取消 \cprotect 的外部宏限制。 71

改用 indentfirst 宏包处理缩进的问题。 67

放弃使用放缩字体大小的方式，而只采用调整间距的方式与西文等宽字体对齐。并且只适用于与抄录环境下。 . 63

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。 . . . 58

\c_xeCJK_space_skip_tl: 字间空格考虑 \spaceskip 不为零的情况。 14

LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。 65

\xeCJK@fix@penalty: 采用通过不修改原语 \/ 的方式对修复倾斜校正。 69

\xeCJK_fallback_loop:Nn: 调整备用字体的循环方式。 . . 49

\xeCJK_glyph_if_exist:N: 改进 fontspec 宏包中定义的 \font_glyph_if_exist:NnTF。 14

\xeCJK_hook_for_ulem:: 简化对 ulem 宏包的兼容补丁。 . . 72

\xeCJK_visible_space_fallback:: 调整 fontspec 的后备可视空格符号，以便于使用时对齐。 64

\xeCJKVerbAddon: 新增 \xeCJKVerbAddon 用于抄录环境中的间距调整。 63

v3.1.1

General: 不再依赖 xpatch 宏包。 1

增加 NewLineCS 和 EnvCS 选项。 35

增加小宏包 xeCJKfntef，用于处理下划线的问题。 72

对于与 xltxtra 的冲突给出错误警告。 68

__xeCJK_check_single_space:NN: CheckSingle 支持段末“汉字 + 汉字 + 空格 + 汉字/标点”的形式。 35

__xeCJK_set_char_class_eq:nn: 交换参数的顺序。 22

__xeCJK_set_verb_exspace:: 调整间距的计算方法。 . . . 64

\fontfamily: 修改主要 CJK 字体族的自动更新方式。 68

CheckFullRight: 处理全角右标点之后的断行问题。 32

PlainEquation: 增加 PlainEquation 选项。 36

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。 36

\xeCJK_check_single:NNw: 改进定义，减少使用 peek 函数的次数。 34

\xeCJK_fallback_loop:Nn: 修正备用字体的不正确警告。 . 49

\xeCJK_hook_for_ulem:: 完全处理下划线里的标点符号的有关问题。 72

\xeCJK_peek_catcode_ignore_spaces:NTF: 新增有省略空格标识的 peek 函数。 15

\xeCJK_save_class:nn: 使用 \xeCJK_save_class:nn 保存 XeTeX 预定义的字符类别。 17

\xeCJK_set_char_class:nnn: 在文档中设置字符类别时不重复设置 \catcode。 22

\xeCJKnobreak: 增加 \nobreak 的 xeCJK 版本。 33

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

\@@italiccorr 2740, 2748, 2754, 2760, 2763, 2774

\@ifpackagelater 19, 22, 2702, 2712

\@ifpackageloaded 2903

\@onefilewithoptions 2895

\@onlypreamble 1094, 1681, 1688, 2328, 2329, 2330, 2331, 2332, 2362

\@pkgextension 51

\[. 2637

\\ 5, 6, 7, 13, 14, 15, 16, 231, 232, 1130, 1182, 1183, 1664, 1665, 1666, 1678, 1679, 1757, 2300, 2389, 2431, 2432, 2433, 2434, 2615, 2658, 2659, 2697, 3166

__xeCJK_Boundary_and_FullLeft_glue:N 746, 754, 2968

__xeCJK_CJK_and_FullLeft_glue:N 736, 741, 2964

__xeCJK_CJK_and_FullRight_glue:N . 787, 793, 793, 803, 2966

__xeCJK_Default_and_FullLeft_glue:N 722, 731, 2960

__xeCJK_Default_and_FullRight_glue:N 762, 775, 793, 803, 2962

__xeCJK_add_fake_bold:n 2006, 2006, 2055, 2074

__xeCJK_add_fake_slant:n 2006, 2011, 2066

__xeCJK_add_font:nV 2032, 2034

__xeCJK_add_font:nn 2000, 2000, 2004

__xeCJK_add_font:nv 2022

__xeCJK_add_font:nx 2082

__xeCJK_add_font_if_new:nn 2000, 2002, 2005, 2076

__xeCJK_add_font_if_new:nv 2026

__xeCJK_add_font_if_new:nx 2084

__xeCJK_add_special_punct:nn . 1207, 1210, 1213, 1217, 1234

__xeCJK_after_end_preamble:n 52, 59, 1189, 2777, 2831, 2842

__xeCJK_after_preamble:n 52, 57, 2503, 2814

__xeCJK_at_end_preamble:n 52, 55, 2410, 2708, 2867

__xeCJK_aux_r:n 2823, 2824

__xeCJK_block_select_font:n 2212, 2216, 2216

__xeCJK_calc_kerning_margin:NN 1470, 1490, 1490

__xeCJK_calc_kerning_margin:nNN . . 1503, 1512, 1557, 1557

__xeCJK_check_family:V 1950, 2136

__xeCJK_check_family:n 1988, 1988, 1997

__xeCJK_check_num_range:nnNN 345, 360, 368, 368, 2470

__xeCJK_check_single_save:N 932, 939, 956, 965, 990, 991, 1001, 1006, 1007, 1015, 1018, 1027, 1030

__xeCJK_check_single_space:NN 964, 996, 996

__xeCJK_class_csname:n 208, 211, 214, 219, 222, 227, 227, 228, 312, 314, 319, 348, 1118, 1136, 2791

__xeCJK_clear_Boundary_and_CJK_toks:n 439, 440

__xeCJK_copy_family:nn 2163, 2163, 2182

__xeCJK_copy_family:xx 2224

__xeCJK_copy_sub_family:n 2126, 2134

__xeCJK_dim_max:nn 2706, 2714, 2718

__xeCJK_dim_min:nn 2707, 2715, 2719

__xeCJK_document_left_hook: 69, 75

__xeCJK_document_right_hook: 71, 77

__xeCJK_error:n 36, 37, 1188

__xeCJK_error:nx 36, 38, 209, 220, 1124, 1658, 1686, 2609, 2703

__xeCJK_fallback_save_CJKsymbol:N 1720, 1727, 1735, 1744, 1750
__xeCJK_family_csname:n 1976, 2150, 2227, 2235, 2235, 2253, 2666
__xeCJK_family_nfss_csname:n 1993, 2178, 2179, 2235, 2236, 2237, 2241, 2251
__xeCJK_family_unknown_warning:n 2282, 2288, 2296
__xeCJK_family_unknown_warning:x 2272, 2286
__xeCJK_family_use:x 2191, 2200, 2231, 2235, 2237, 2270, 2285
__xeCJK_fill_newline: 3166, 3178, 3178
__xeCJK_font_csname:n .. 2184, 2185, 2197, 2199, 2218, 2221
__xeCJK_get_map_font:n 2082, 2084, 2087, 2087, 2091
__xeCJK_get_sub_features:Vn 1778
__xeCJK_get_sub_features:nn 1841, 1848, 1848, 1875
__xeCJK_get_sub_features:wn 1848, 1854, 1868
__xeCJK_gobble_CJKfamily: 2279, 2279, 2837
__xeCJK_gobble_CJKfamily:wn 2280, 2281
__xeCJK_gset_family_cs:x 1951, 1974, 1974
__xeCJK_gset_family_nfss_cs:xx ... 1984, 2157, 2235, 2238
__xeCJK_gset_punct_dim:nnnnx 1267, 1281, 1612
__xeCJK_gset_punct_dim:nnnx 1267, 1279, 1383, 1458, 1578, 1579, 1580, 1585, 1595, 1597, 1636
__xeCJK_gset_punct_dim:nnx 1267, 1277, 1604, 1606
__xeCJK_if_font_select:n 2040
__xeCJK_if_font_select:nF 2025, 2031, 2033, 2083
__xeCJK_if_font_select:nTF 2020, 2040
__xeCJK_if_font_select_p:n 2040, 2062, 2064, 2072
__xeCJK_if_it_or_sl:n 2045
__xeCJK_if_it_or_sl:nT 2068
__xeCJK_if_it_or_sl:nTF 2023, 2040
__xeCJK_if_it_or_sl_p:n 2040, 2060, 2071
__xeCJK_if_map_font_select:n 2089
__xeCJK_if_map_font_select:nT 2080
__xeCJK_if_map_font_select:nTF 2089
__xeCJK_if_map_font_select_p:n 2089
__xeCJK_info:nx 36, 42
__xeCJK_info:nxx 36, 43, 2209
__xeCJK_int_until_do:wn 135, 136, 137, 138
__xeCJK_kerning_width_or_ratio:nNN 1508, 1509, 1527, 1527
__xeCJK_make_under_symbol:n 3118, 3134, 3141, 3141
__xeCJK_map_it_sl:n 2078
__xeCJK_map_it_sl:nn 2068, 2078
__xeCJK_margin_width_or_ratio:n .. 1424, 1425, 1445, 1445
__xeCJK_msg_family_map:n 1999, 2300, 2311
__xeCJK_msg_new:nn 36, 36, 229, 1128, 1180, 1662, 1676, 1753, 1998, 2215, 2298, 2387, 2429, 2613, 2656, 2695
__xeCJK_original_kerning_margin:NN 1471, 1478, 1478, 1541
__xeCJK_parse_features: 1948, 2016, 2016
__xeCJK_parse_features:nn 2016, 2017, 2018
__xeCJK_parse_sub_class_features:n 2375, 2392, 2392
__xeCJK_pass_features: 1949, 2094, 2094
__xeCJK_peek_after_do:w 169, 180
__xeCJK_peek_catcode_false:w 143, 162
__xeCJK_peek_catcode_ignore_spaces_branches:w 146, 148, 153
__xeCJK_peek_catcode_true:w 142, 159
__xeCJK_peek_ignore_spaces_branches:w ... 171, 173, 177
__xeCJK_prop_put_aux:n 1960, 1970
__xeCJK_prop_put_aux:nn 1960, 1972
__xeCJK_punct_break: 829, 1202, 1204
__xeCJK_punct_dim_csname:nn 1267, 1267, 1274, 1278
__xeCJK_punct_dim_csname:nnn 1267, 1269, 1276, 1280, 1574, 1633
__xeCJK_punct_dim_csname:nnnn 1267, 1270, 1271, 1282
__xeCJK_punct_glue:NN . 675, 675, 693, 703, 710, 717, 732, 742, 757, 799, 870, 3018, 3026, 3035, 3049, 3068, 3085, 3094
__xeCJK_punct_hskip:n 677, 683, 2958
__xeCJK_punct_if_long:NT 1608
__xeCJK_punct_if_long:NTF 795, 1253, 1584, 3043
__xeCJK_punct_if_long_p:N 1253
__xeCJK_punct_if_middle:NT 768, 780, 789, 3046
__xeCJK_punct_if_middle:NTF 689, 699, 798, 1253, 1374, 1390, 1411, 3063
__xeCJK_punct_if_middle_p:N 1253
__xeCJK_punct_if_mixed_width:NTF 1253, 1377, 1423
__xeCJK_punct_if_mixed_width_p:N 1253
__xeCJK_punct_if_right:N 1253
__xeCJK_punct_if_right:NTF ... 1253, 1483, 1486, 1563, 1566
__xeCJK_punct_if_right_p:N 1253
__xeCJK_punct_kern:NN 684, 684, 809, 818, 827, 836
__xeCJK_punct_min_bound:NN 1496, 1499, 1543, 1543
__xeCJK_punct_rule:NN 668, 668, 691, 701, 709, 715, 728, 738, 751, 769, 781, 790, 868, 3066, 3082, 3091
__xeCJK_punct_width_or_ratio:nN 1375, 1378, 1379, 1432, 1432
__xeCJK_remove_duplicate_keys:N 1945, 1957, 1957
__xeCJK_save_CJK_class:n 309, 318, 1173
__xeCJK_save_FullRight_check: 848, 858
__xeCJK_save_FullRight_symbol:N 849, 859, 875
__xeCJK_save_family_info: 1952, 2110, 2110
__xeCJK_save_under_dot_CJKsymbol:N 3119, 3124, 3135, 3157
__xeCJK_set_char_catcode:n 349, 350, 353
__xeCJK_set_char_class:nnn 358, 358, 382
__xeCJK_set_char_class_aux:Nnw 325, 334, 339, 2463
__xeCJK_set_char_class_catcode:nnn 343, 343, 380
__xeCJK_set_char_class_eq:nn 383, 383, 2485, 2486, 2487, 2488, 2489
__xeCJK_set_fake_font:nn 2038, 2052, 2052
__xeCJK_set_family_initial: 1914, 1914, 1939
__xeCJK_set_others_toks:n 2777, 2785, 2789
__xeCJK_set_special_punct:nn . 1206, 1209, 1212, 1217, 1223
__xeCJK_set_sub_block_family: 1953, 2121, 2121
__xeCJK_set_sub_class_toks:nn 1139, 1145, 1145
__xeCJK_set_verb_exspace: 2481, 2504, 2504
__xeCJK_set_visible_space_size:n 2542, 2546
__xeCJK_space_or_xecglue: ... 476, 586, 591, 598, 621, 634
__xeCJK_special_punct_seq:n 1219, 1222, 1225, 1227, 1231, 1238, 1241, 1250
__xeCJK_special_punct_tl:nN 1220, 1226, 1230, 1240, 1249, 1263
__xeCJK_sub_restore_or_cancel:n 1114, 1114, 1127
__xeCJK_sub_restore_or_cancel:x 1101, 1110
__xeCJK_sub_special_punct:nn . 1208, 1211, 1214, 1217, 1245
__xeCJK_swap_cs_aux:w 96, 98, 99
__xeCJK_switch_font:nn 1154, 1159, 1168, 1170, 1177, 2205, 2205, 3011
__xeCJK_tl_remove_outer_braces:w 85, 86, 89
__xeCJK_token_value_charcode:w 186, 187
__xeCJK_ulem_Boundary_and_FullLeft_glue:N 2968, 3021, 3021
__xeCJK_ulem_CJK_and_CJK:N 2973, 2992, 2992
__xeCJK_ulem_CJK_and_FullLeft_glue:N .. 2964, 3029, 3029
__xeCJK_ulem_CJK_and_FullRight_glue:N . 2966, 3055, 3055

__xeCJK_ulem_Default_and_FullLeft_glue:N 2960, 3014, 3014
__xeCJK_ulem_Default_and_FullRight_glue:N 2962, 3039, 3039, 3058
__xeCJK_ulem_FullLeft_and_CJK: 2970, 3075, 3075
__xeCJK_ulem_FullLeft_and_Default: 2969, 3061, 3061, 3077
__xeCJK_ulem_FullRight_and_CJK: 2972, 3089, 3089
__xeCJK_ulem_FullRight_and_Default: ... 2971, 3080, 3080
__xeCJK_ulem_between_CJK_blocks:nnN 2981, 2983, 2987, 3005, 3005
__xeCJK_ulem_ccglue: 2995, 3008, 3034, 3044, 3095, 3100, 3106
__xeCJK_ulem_class_group_begin: 2996, 2999, 2999, 3037, 3059, 3078, 3098
__xeCJK_ulem_glue:n 2937, 2939, 3100, 3100
__xeCJK_ulem_hskip:n 2958, 3100, 3104, 3108
__xeCJK_ulem_initial: 2921, 2956, 2956
__xeCJK_ulem_leader_type: 2916, 2954, 2955
__xeCJK_ulem_skip_punct_begin: 2918, 2951, 2951, 3017, 3024, 3033, 3042
__xeCJK_ulem_skip_punct_end: 2919, 2951, 2953, 3071, 3086, 3096
__xeCJK_under_CJKsymbol:N 3120, 3136, 3155, 3155
__xeCJK_update_cs_case_tl:NNnn 885, 891, 896, 900
__xeCJK_update_family:n 1980, 2725, 2737
__xeCJK_update_inline_env_case_tl: 1059, 1068, 1074, 1078, 1078
__xeCJK_use_punct_dim:nn 1267, 1273, 1393, 1402, 1441, 1516, 1517, 1568, 1569, 1607
__xeCJK_use_punct_dim:nnn 671, 679, 680, 685, 1267, 1275, 1354, 1357, 1473, 1474, 1482, 1485, 1548, 1549, 1553, 1554, 1562, 1565, 1601, 1602
__xeCJK_warning:n 39, 2384, 2413
__xeCJK_warning:nx 36, 40, 1671, 2293, 2652, 2653, 2654
__xeCJK_warning:nxx 41, 1749, 1994

\␣ 1131, 1182, 2615, 2659, 2660, 2924, 2931, 3102

A

\addCJKfontfeature 2368, 2389
\addCJKfontfeatures 6, 2359, 2363, 2368
\addto@hook 2908
\AfterEndPreamble 65
\AfterPreamble 64
KaiMingPunct 4
KaiMingPunct+ 4
KaiMingPunct- 4
FallBack 6
\AllowBreakBetweenPuncts 1198
\AssignTemplateKeys 1347
\AtBeginDocument 68
\AtEndOfPackage 381, 2534
\AtEndPreamble 63
\AutoFakeBold 1792, 1893
\AutoFakeSlant 1792, 1893
\AutoFallBack 1713

B

\begin 2638
\bfdefault 2455
\BoldFeatures 1879
\BoldFont 1879
\BoldItalicFeatures 1879
\BoldItalicFont 1879
\BoldSlantedFeatures 1879
\BoldSlantedFont 1879
\bool_gset_false:N 2569
\bool_gset_true:N 1798, 1806, 2572, 2577
\bool_if:cT 2126, 2127
\bool_if:NF 1098, 1121, 2482, 2911
\bool_if:NT 1107, 1453, 2055, 2426, 2649, 2880, 2892, 3116, 3121, 3123, 3126
\bool_if:nT 518, 628, 2059, 2069, 2844
\bool_if:NTF 539, 620, 624, 963, 972, 984, 989, 1362, 1397, 1407, 1469, 1495, 1498, 2057, 2915
\bool_if:nTF 88, 370, 460, 467, 507, 511, 526, 534, 949, 978, 1462, 1614
\bool_if_p:n 2048
\bool_new:N 166, 603, 1095, 1759, 1788, 1789, 2501, 2583, 2942, 2949
\bool_set_eq:NN 1930, 1931
\bool_set_false:c 1827, 1828, 1836, 1839, 1927, 1928
\bool_set_false:N 144, 590, 1109, 1897, 1906, 2948
\bool_set_true:c 1835, 1840
\bool_set_true:N 151, 585, 595, 1100, 1900, 1909, 2484, 2913, 2946
\Boundary 234
\box_new:N 30, 3129
\box_use:N 3150, 3158
\box_wd:N 3149

C

\c__xeCJK_CJK_chars_clist 242, 268, 403, 2458
\c__xeCJK_config_ext_tl 2894, 2896
\c__xeCJK_encoding_tl 2243, 2453, 2455, 2648, 2648
\c__xeCJK_FullLeft_chars_clist 242, 247, 398, 2458
\c__xeCJK_FullRight_chars_clist 242, 254, 399, 2460
\c__xeCJK_HalfLeft_chars_clist 242, 242, 396
\c__xeCJK_HalfRight_chars_clist 242, 244, 397
\c__xeCJK_left_tl 693, 703, 721, 728, 732, 735, 738, 742, 745, 751, 757, 769, 781, 790, 807, 825, 1196, 1196, 1358, 1474, 1487, 1548, 1553, 1564, 1579, 1595, 1601, 3018, 3026, 3035, 3065, 3068
\c__xeCJK_math_family_tl 2451, 2453, 2455
\c__xeCJK_math_tl .. 2438, 2439, 2442, 2443, 2452, 2454, 2456
\c__xeCJK_NormalSpace_chars_clist 242, 246, 404
\c__xeCJK_package_ext_tl 48, 51
\c__xeCJK_punct_style_plain_tl 1576, 1635, 1653, 1661
\c__xeCJK_right_tl 691, 701, 709, 710, 715, 717, 761, 774, 786, 799, 816, 834, 867, 868, 870, 1196, 1197, 1358, 1473, 1484, 1549, 1554, 1567, 1580, 1597, 1602, 3049, 3066, 3082, 3085, 3091, 3094
\c_catcode_other_token 946, 961
\c_eleven 911
\c_empty_tl 1854, 1871, 1872
\c_four 117, 2782
\c_group_begin_token 442, 3168, 3169, 3182
\c_group_end_token 443, 3174, 3175, 3181
\c_math_toggle_token 618, 1013
\c_max_dim 1287, 1293, 1304, 1307, 1309, 1311, 1388, 1413, 1434, 1438, 1447, 1502, 1529, 1537
\c_one 201, 235, 756, 1596, 1764, 2352, 2782, 3025
\c_one_fp 1288, 1294, 1312
\c_seven 2464
\c_space_tl 591
\c_space_token 150, 175
\c_ten 514, 529

\c_ten_thousand 45
\c_three 116, 237, 1283, 1284, 1315, 1598, 2780, 3110
\c_two 115, 236, 680, 1394, 1418, 2518, 2543, 2551, 2558
\c_two_hundred_fifty_five 238
\c_two_hundred_fifty_six 2886
\c_xeCJK_math_fam_int 2456, 2464
\c_xeCJK_space_skip_tl 111, 111, 463, 521
\c_zero 44, 202, 234, 2479, 2744
\c_zero_dim 672, 673, 1301, 1313, 1578, 1579, 1580, 1585, 1619, 1636, 2497, 2516, 2622, 3144
\c_zero_fp 1308, 1310, 1437, 1505, 1532
\c_zero_skip 113
\char_set_catcode:nn 2860
\char_set_catcode_ignore:n 203
\char_set_catcode_letter:n 2858
\char_set_catcode_other:n 349
\char_value_catcode:n 2860
\CheckFullRight 841
\CheckSingle 925
\CJK 234
\CJK@@UL 2944, 2944, 2947
\CJK@hundredmillion 2890
\CJK@ifundefined 2851, 2852
\CJK@postUnderdot 3125
\CJK@preUnderdot 3117
\CJK@tenthousand 2889
\CJK@UL 2944, 2944, 2945
\CJK@underdotBox 3133, 3150
\CJK@underdotSkip 3118
\CJK@UnicodeEnc 2882, 2888
\CJKaddEncHook 2883
\CJKecglue 472, 475, 480, 495, 516, 532, 575, 579, 586, 596, 598, 621, 2493, 2495, 2496, 2934, 2938
xCJKecglue 3
\CJKfamily 5, 2256, 2256, 2280, 2338, 2356, 2425, 2729, 2730, 2731, 2732
\CJKfamilydefault 6, 2227, 2228, 2278, 2401, 2415, 2423, 2425, 2445, 2446, 2662, 2665, 2666, 2732
\CJKfilltwosides 3162
\CJKfixedspacing 2502
\CJKfontspec 6, 2333, 2341
\CJKglue 509, 539, 566, 570, 640, 711, 742, 796, 2491, 2927, 2936, 3165
\CJKmath 2436
\CJKnospace 2680, 2681
\CJKnumber 2584
\CJKpunctsymbol 729, 739, 752, 811, 830, 864, 2618, 2619
\CJKrmdefault 6, 2304, 2315, 2322, 2419, 2662, 2662, 2665, 2729
\CJKsetecglue 2678, 2678, 2679
\CJKsfdefault 6, 2305, 2316, 2325, 2420, 2662, 2663, 2730
\CJKspace 604, 2680, 2680
\CJKsymbol 452, 503, 640, 642, 644, 1158, 1159, 1718, 1720, 1721, 1726, 1727, 2618, 2618, 2997, 3012, 3119, 3120, 3124, 3135, 3136
\CJKttdefault 6, 2306, 2317, 2327, 2421, 2662, 2664, 2731
\CJKunderanysymbol 3130, 3130
\CJKunderdot 3114, 3114
\clist_clear:c 1921
\clist_clear:N 1851, 1858, 1923, 1961
\clist_concat:NNN 1943, 2379, 2457, 2459
\clist_const:Nn 242, 244, 246, 247, 254, 268
\clist_count:N 1764
\clist_gconcat:ccN 328
\clist_gconcat:NNN 341
\clist_gset:Nn 1218, 2361
\clist_if_empty:cF 2098
\clist_if_in:NnT 2394
\clist_map_break: 2406
\clist_map_function:nN 318
\clist_map_inline:cn 386
\clist_map_inline:Nn 330, 1221, 1258, 1775, 2461
\clist_map_inline:nn 444, 456, 482, 647, 663, 665, 1063, 1072, 1116, 1174
\clist_new:c 212, 223
\clist_new:N 34, 35, 1217, 2359
\clist_put_left:cn 1779
\clist_put_left:cx 2008, 2013
\clist_put_right:Nn 1965, 1966, 2105
\clist_put_right:Nx 2100
\clist_remove_all:Nn 2114, 2146, 2396
\clist_set:Nn 327, 1870, 1941, 2373
\clist_set:Nx 1763, 2404
\clist_set_eq:cN 1865
\cs:w 104, 2846
\cs_end: 104, 1263, 2846
\cs_generate_variant:Nn 317, 341, 342, 410, 420, 426, 873, 1010, 1041, 1127, 1144, 1193, 1625, 1643, 1760, 1787, 1875, 1876, 1877, 1956, 1969, 1997, 2004, 2005, 2051, 2182, 2248, 2296, 2358, 2386, 2409
\cs_gset_eq:cc 2177
\cs_gset_eq:NN 103, 2537, 2714, 2715, 3119, 3135
\cs_gset_nopar:Npx 73, 2821
\cs_gset_protected_nopar:cpx 1976, 2150, 2241
\cs_if_eq:NNF 846, 930, 1718
\cs_if_eq:NNT 856, 938, 1726
\cs_if_eq:NNTF 2864, 3102
\cs_if_exist:cTF 2226, 2251
\cs_if_exist:NF 636, 1186, 2851
\cs_if_exist_p:N 2845
\cs_if_exist_use:c 2888
\cs_if_exist_use:cF 2188, 2197, 2218, 2530
\cs_if_exist_use:cTF 1364, 2253
\cs_if_free:NTF 2852
\cs_new:Npn 84, 86
\cs_new_eq:cN 228
\cs_new_eq:NN 92, 93, 380, 442, 443, 683, 803, 2204, 2323, 2368, 2502, 2679, 2706, 2707, 2737, 2739, 2823
\cs_new_nopar:Npn 80, 183, 185, 187, 227, 319, 411, 864, 1219, 1220, 1267, 1269, 1271, 1273, 1275, 1432, 1445, 1478, 1490, 1527, 1543, 1557, 1626, 2087, 2185, 2235, 2236, 2237, 2311, 2618, 2619
\cs_new_protected:Npn 55, 57, 59, 94, 135, 137, 139, 167, 876, 900, 1078, 3141
\cs_new_protected_nopar:Npn 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 69, 71, 82, 101, 121, 148, 173, 206, 217, 309, 325, 343, 358, 368, 383, 408, 413, 415, 421, 427, 432, 438, 440, 505, 551, 557, 616, 640, 661, 668, 675, 684, 687, 697, 707, 713, 719, 731, 733, 741, 743, 754, 759, 772, 784, 793, 804, 813, 823, 832, 865, 874, 944, 959, 996, 1011, 1020, 1033, 1114, 1134, 1145, 1223, 1234, 1245, 1277, 1279, 1281, 1352, 1456, 1572, 1593, 1631, 1732, 1738, 1768, 1818, 1848, 1868, 1914, 1936, 1957, 1970, 1972, 1974, 1988, 2000, 2002, 2006, 2011, 2016, 2018, 2052,

2078, 2094, 2110, 2121, 2134, 2163, 2186, 2195, 2205, 2216, 2238, 2279, 2282, 2288, 2346, 2369, 2392, 2440, 2468, 2504, 2528, 2539, 2546, 2742, 2789, 2909, 2951, 2953, 2956, 2992, 2999, 3005, 3014, 3021, 3029, 3039, 3055, 3061, 3075, 3080, 3089, 3100, 3106, 3108, 3155, 3178
\cs_new_protected_nopar:Npx 2338
\cs_set:cpn 2884
\cs_set_eq:NN 96, 97, 98, 141, 349, 350, 382, 586, 591, 598, 848, 849, 850, 851, 858, 859, 932, 933, 939, 1202, 1204, 1720, 1721, 1727, 1979, 2280, 2495, 2496, 2852, 2916, 2924, 2925, 2926, 2931, 2932, 2933, 2944, 2954, 2958, 2959, 2961, 2963, 2965, 2967, 2969, 2970, 2971, 2972, 2973, 3120, 3124, 3136, 3165, 3166
\cs_set_nopar:Npn 2824
\cs_set_protected:Npn 2883
\cs_set_protected_nopar:Npn 570, 579, 596, 2491, 2493, 2885, 2936, 2938
\cs_to_str:N 2337
\cs_undefine:c 1226, 1249, 1993
\cs_undefine:N 99, 2718, 2719
\curr@fontshape 2506, 2514, 2525, 2530, 2541
\CurrentOption 2901

D

\DeclareExpandableDocumentCommand 2281
\DeclareInstance 1673
\DeclareObjectType 1283
\DeclareOption 2901
\DeclareSymbolFont 2452
\DeclareTemplateCode 1315
\DeclareTemplateInterface 1284
\Default 234
\defaultCJKfontfeatures 6, 2359, 2360, 2362, 2645
\dim_compare:nNnF 2551
\dim_compare:nNnTF . 1388, 1413, 1434, 1447, 1502, 1529, 1537
\dim_const:cn 554
\dim_eval:n 680, 1460, 1480, 1559, 1607, 2555
\dim_gset:cn 554
\dim_if_exist:cTF 553
\dim_max:nn . . . 1385, 1398, 1492, 1499, 1545, 2516, 2706, 2714
\dim_min:nn 1398, 1408, 1454, 1547, 1552, 2707, 2715
\dim_new:N 31, 3177
\dim_ratio:nn 2558
\dim_set:Nn 1360, 1599, 3164
\dim_use:N 1605, 1620, 1628, 2519, 2543
\document 73, 76

E

xeCJKactive 3
\EditInstance 1685
\else: . 109, 130, 132, 155, 160, 179, 192, 549, 1256, 1264, 2049
\EmboldenFactor 1792
\end 2638
\EnvCS 1048
\etex_currentgrouplevel:D 2479
\etex_dimexpr:D 1417, 1441, 1451, 1515, 1541
\etex_fontcharwd:D 1605, 2519
\etex_iffontchar:D 108
\etex_lastnodetype:D 514, 529, 756, 911, 3025
\etex_numexpr:D 2886
NewLineCS 3
NewLineCS+ 3
NewLineCS- 3
\exp_after:wN 103, 104, 123, 124, 138, 152, 153, 158, 159, 161, 162, 176, 177, 180, 186, 953, 955, 1854, 2542, 2543, 2766, 2767, 2768, 2769, 2846
\exp_args:Nc 211, 1260, 2356
\exp_args:No 952
\exp_args:NV 83
\exp_not:c 314, 650
\exp_not:N 157, 314, 636, 1984, 2154, 2155, 2158, 2243, 2244, 2245, 2516
\exp_not:n 418, 424, 1980, 1981, 2514
\exp_not:o 76, 2821
\exp_not:V 436, 1982, 1983, 2405
\exp_not:v 2088, 2101
\exp_stop_f: 108, 128, 131, 191, 1629

F

\f@baselineskip 2561
\f@encoding 2243, 2548
\f@family 2244, 2549, 2724
\f@series 2185
\f@shape 2185
\f@size 2185, 2506, 2514, 2525, 2530, 2541, 2557
\FallBack 1878
\familydefault 2417, 2732
\fi: 109, 132, 133, 138, 163, 164, 181, 192, 549, 1256, 1264, 2049
\file_input:n 2859
\fix@penalty 2739, 2741
\fontfamily 2722, 2722
\fontsize 2553
\fontspec_set_family:Nnn 1981
\fontspec_visible_space: 2536
\fontspec_visible_space_fallback: 2537
\footnote 2639
\footnotemark 2639
\fp_compare:nNnTF 1437, 1505, 1532
\fp_gset:Nn 1799, 1807
\fp_new:N 1790, 1791
\fp_set:Nn 1901, 1910
\fp_set_eq:NN 1932, 1933
\fp_use:c 1440, 1450, 1534
\fp_use:N 1416, 1514, 1539, 2009, 2014
\FullLeft 234
\FullRight 234

G

\g__xeCJK_after_end_preamble_hook_tl 54, 60, 65, 72
\g__xeCJK_after_preamble_hook_tl 53, 58, 64, 68
\g__xeCJK_at_end_preamble_hook_tl 52, 56, 63, 70
\g__xeCJK_auto_fake_bold_bool 1788, 1795, 1798, 1930
\g__xeCJK_auto_fake_slant_bool 1789, 1803, 1806, 1931
\g__xeCJK_base_class_seq 302, 302, 303, 1147
\g__xeCJK_check_single_CJK_case_tl . . . 313, 316, 999, 1004
\g__xeCJK_CJK_class_prop 302, 308, 312
\g__xeCJK_CJK_class_seq 302, 307, 311, 439, 2792
\g__xeCJK_CJK_sub_class_seq 1087, 1087, 1163, 1172, 2113, 2375, 2974, 2976
\g__xeCJK_class_seq 204, 204, 213, 224, 303
\g__xeCJK_config_bool 2569, 2572, 2577, 2583, 2892
\g__xeCJK_config_name_tl 2573, 2578, 2582, 2896
\g__xeCJK_default_features_clist 1944, 2359, 2361
\g__xeCJK_embolden_factor_fp . . 1790, 1799, 1810, 1903, 1932
\g__xeCJK_family_font_name_prop 1756, 1772, 1990, 2107, 2108, 2115, 2137, 2140, 2170, 2371, 2397, 2400, 2412

\g__xeCJK_family_font_options_prop 2107, 2109, 2117, 2143, 2147, 2171, 2376
\g__xeCJK_family_int 1916, 1935, 2352
\g__xeCJK_family_name_prop 1992, 2107, 2107, 2154, 2169, 2240, 2449
\g__xeCJK_features_id_prop 1879, 1879, 1880, 1881, 1882, 1883, 1884, 1885, 1918, 2017, 2096
\g__xeCJK_indent_bool 2587, 2649
\g__xeCJK_last_punct_tl 686, 686, 689, 691, 693, 699, 701, 703, 709, 710, 715, 717, 727, 737, 750, 767, 779, 788, 808, 809, 810, 817, 818, 819, 826, 827, 828, 835, 836, 837, 867, 868, 870, 3063, 3065, 3066, 3068, 3082, 3085, 3091, 3094
\g__xeCJK_math_bool 2426, 2436
\g__xeCJK_new_class_seq 204, 205, 214, 2780, 2784
\g__xeCJK_non_CJK_class_seq 302, 304, 305, 645
\g__xeCJK_number_bool 2586, 2880
\g__xeCJK_punct_style_seq 1610, 1666, 1672, 1675
\g__xeCJK_punct_width_tl 1215, 1367, 1369
\g__xeCJK_slant_factor_fp 1791, 1807, 1811, 1912, 1933
\g__xeCJK_special_punct_clist . 1217, 1217, 1218, 1221, 1258
\g__xeCJK_sub_key_seq 1817, 1817, 1820, 1925
\g__xeCJK_under_symbol_box 3129, 3144, 3158
\g__xeCJK_unknown_family_seq 2290, 2292, 2297
\g__xeCJK_verb_exspace_skip 2492, 2494, 2524, 2527
\g_fontspec_encoding_tl 2548, 2648
\group_align_safe_begin: 145, 170, 968
\group_align_safe_end: 142, 143, 169, 971, 977
\group_begin: 70, 1770, 1938, 1978, 2112, 2508, 3132
\group_end: 70, 1785, 1954, 1985, 2119, 2512, 3138

H

\HalfLeft 239
\HalfRight 239
\hbox_overlap_left:n 3158
\hbox_set:Nn 123, 3133, 3143
\hbox_to_zero:n 3147
CheckFullRight 5
CheckSingle 3
\hskip 2926, 2933

I

\icprotect 2845, 2846
MiddlePunct 4
MiddlePunct+ 4
MiddlePunct- 4
\if_case:w 128, 131
\if_catcode:w 156
\if_cs_exist:w 1263
\if_dim:w 548
\if_int_compare:w 136, 190, 1255
\if_meaning:w 150, 175
\if_predicate:w 2047
\IfBooleanF 2260
\IfBooleanT 323, 1092, 1102, 1111, 2270
\IfBooleanTF 2264
\ifCTEX@fntef 2864
\IfInstanceExistTF 1656, 1670, 1684
\IfNoValueF 2855
\IfNoValueTF 376, 377, 2337
\indentfirst 2584
\InlineEnv 1054
\int_case:nnn 1010
\int_case:non 998, 1003
\int_compare:nNnF 756, 2479, 2779, 3025
\int_compare:nNnTF 347, 1764, 2744, 3110
\int_compare_p:nNn 514, 529
\int_const:cn 222
\int_const:Nn 2456
\int_eval:n 2352
\int_gincr:N 1916
\int_if_even:nF 922
\int_if_even:nTF 913
\int_if_exist:cF 1136
\int_if_exist:cTF 208, 219, 1118
\int_incr:N 355, 365, 915, 2474
\int_max:nn 377
\int_min:nn 376
\int_new:N 27, 28, 29, 1935
\int_set:cn 2791
\int_set:Nn 346, 361, 372, 376, 377, 385
\int_set_eq:NN 373
\int_step_inline:nnnn 2782
\int_to_hexadecimal:n 1757
\int_while_do:nNnn 911
\int_zero:N 910
\ItalicFeatures 1879
\ItalicFont 1879

J

CJKecglue 3
CJKglue 3
CJKmath 3
CJKnumber 3
CJKspace 3

K

\KaiMingPunct 1198
\keys_define:nn 194, 566, 575, 604, 841, 880, 925, 1054, 1085, 1198, 1644, 1713, 1792, 1821, 1887, 1893, 2436, 2565, 2584, 2590, 2650
\keys_set:nn 2620, 2669
\keys_set_known:nVN 1946
\keyval_parse:NNn 1969
\keyval_parse:NNV 1960
\KeyValue 1289, 1290, 1291, 1292, 1295, 1296, 1297, 1298

L

\l__keys_module_tl 2615
\l__peek_search_token 141, 157
\l__xeCJK_add_font_prop 1917, 2001, 2003, 2104
\l__xeCJK_add_min_bound_to_margin_bool 1330, 1453
\l__xeCJK_auto_fake_bold_bool . 1897, 1900, 1930, 2055, 2070
\l__xeCJK_auto_fake_slant_bool 1906, 1909, 1931, 2057
\l__xeCJK_ccglue_skip 571, 574, 2928, 2937, 3107
\l__xeCJK_check_single_cs_case_tl .. 1023, 1032, 1045, 1051
\l__xeCJK_current_coor_tl 2190, 2192, 2199, 2201, 2220, 2232, 2541, 2544
\l__xeCJK_different_align_margin_dim 1340
\l__xeCJK_different_align_ratio_fp 1341
\l__xeCJK_ecglue_skip 522, 580, 597, 602, 2935, 2939
\l__xeCJK_embolden_factor_fp 1901, 1932, 2009
\l__xeCJK_enabled_global_setting_bool .. 1317, 1362, 1464
\l__xeCJK_enabled_kerning_bool 1333, 1469
\l__xeCJK_env_cs_case_tl 1046, 1049, 1052
\l__xeCJK_env_cs_seq 1049

\l__xeCJK_fallback_family_tl 1771, 1773, 1777, 1778, 1780, 1781, 1782, 1783
\l__xeCJK_fallback_first_bool 1759
\l__xeCJK_family_name_tl 1829, 1940, 1950, 1951, 2116, 2118, 2125, 2138, 2144, 2152, 2155
\l__xeCJK_fill_width_dim 3164, 3169, 3177, 3182
\l__xeCJK_fixed_margin_ratio_fp 1325
\l__xeCJK_fixed_margin_width_dim 1324
\l__xeCJK_fixed_punct_ratio_fp 1319
\l__xeCJK_fixed_punct_width_dim 1318
\l__xeCJK_font_features_clist 1943, 1945, 1947
\l__xeCJK_font_name_bf_tl 2032, 2034
\l__xeCJK_font_name_tl 1773, 1774, 1863, 1864, 1942, 1983, 2116
\l__xeCJK_font_options_clist 1941, 1944, 2114, 2118
\l__xeCJK_fontspec_family_tl 1981, 1984
\l__xeCJK_fontspec_options_clist 1923, 1947, 1982, 2100, 2105
\l__xeCJK_inline_env_case_tl 1036, 1080, 1082, 1084
\l__xeCJK_inline_env_seq 1058, 1065, 1066, 1073, 1077, 1081
\l__xeCJK_kerning_margin_minimum_dim 1344, 1493
\l__xeCJK_kerning_margin_ratio_fp 1343, 1539
\l__xeCJK_kerning_margin_width_dim 1342, 1537, 1538
\l__xeCJK_kerning_total_ratio_fp 1336, 1505, 1514
\l__xeCJK_kerning_total_width_dim 1335, 1502, 1503
\l__xeCJK_margin_minimum_dim 1332, 1386
\l__xeCJK_middle_margin_ratio_fp 1329, 1416
\l__xeCJK_middle_margin_width_dim 1328, 1413, 1414
\l__xeCJK_middle_punct_ratio_fp 1323
\l__xeCJK_middle_punct_width_dim 1322
\l__xeCJK_min_bound_to_kerning_bool 1334, 1495
\l__xeCJK_mixed_margin_ratio_fp 1327
\l__xeCJK_mixed_margin_width_dim 1326
\l__xeCJK_mixed_punct_ratio_fp 1321
\l__xeCJK_mixed_punct_width_dim 1320
\l__xeCJK_new_line_cs_case_tl 1043, 1046, 1052
\l__xeCJK_new_line_cs_seq 1043
\l__xeCJK_no_break_cs_case_tl 869, 907
\l__xeCJK_no_break_cs_seq 907
\l__xeCJK_optimize_kerning_bool 1337, 1498
\l__xeCJK_optimize_margin_bool 1331, 1397, 1407
\l__xeCJK_peek_ignore_spaces_bool 144, 151, 166, 620, 624, 963, 972, 984, 989
\l__xeCJK_plain_equation_bool 980, 1086
\l__xeCJK_reserve_space_bool 539, 606, 631
\l__xeCJK_same_align_margin_dim 1338
\l__xeCJK_same_align_ratio_fp 1339
\l__xeCJK_slant_factor_fp 1910, 1933, 2014
\l__xeCJK_sub_cancel_bool 1095, 1098, 1100, 1107, 1109, 1121
\l__xeCJK_sub_family_tl 2125, 2129, 2136, 2141, 2148, 2150, 2158
\l__xeCJK_sub_font_name_tl 1855, 1857, 1861, 1862, 1863, 1864, 1866, 1872, 1873
\l__xeCJK_sub_font_options_clist .. 1851, 1858, 1865, 1870
\l__xeCJK_sub_key_seq 1832, 1924, 2123
\l__xeCJK_tmp_box 30, 123, 3143, 3149
\l__xeCJK_tmp_dim 31, 1360, 1388, 1393, 1401, 1599, 1607, 1620
\l__xeCJK_tmp_prop 33, 1959, 1962, 1971, 1973
\l__xeCJK_tmp_skip 32, 916, 918, 923
\l__xeCJK_tmpa_clist 34, 327, 329, 330, 1763, 1764, 1765, 2144, 2146, 2148, 2373, 2380, 2382, 2394, 2396, 2404, 2405, 2458, 2460, 2461
\l__xeCJK_tmpa_int 27, 228, 345, 351, 353, 354, 355, 360, 362, 364, 365, 385, 387, 910, 913, 915, 922, 2470, 2471, 2473, 2474
\l__xeCJK_tmpa_tl 25, 434, 435, 436, 1354, 1398, 1408, 1418, 1451, 1454, 1850, 1852, 1854, 1857, 1861, 1990, 1992, 1994, 2138, 2141, 2155, 2158, 2174, 2175, 2337, 2338, 2339, 2348, 2349, 2353, 2356, 2371, 2382, 2402, 2443, 2446, 2449
\l__xeCJK_tmpb_clist 35, 2377, 2380
\l__xeCJK_tmpb_int 28, 345, 351, 360, 362, 2470, 2471
\l__xeCJK_tmpb_tl 26, 1355, 1398, 1401, 1418, 1454, 2351, 2353, 2354, 2398, 2401, 2402, 2405, 2449, 2451
\l__xeCJK_tmpc_int 29, 346, 348, 354, 361, 364
\l__xeCJK_ulem_hook_used_bool 2911, 2913, 2942, 3116, 3121, 3123, 3126
\l__xeCJK_ulem_skip_punct_bool 2915, 2946, 2948, 2949
\l__xeCJK_verb_addon_bool 2482, 2484, 2501
\l__xeCJK_xecglue_bool 462, 520, 585, 590, 595, 603
\l_keys_choice_tl 1795, 1803
\l_keys_key_tl 2609, 2610, 2652, 2653, 2654
\l_keys_value_tl 1656, 1657, 1658, 1799, 1807, 1901, 1910, 2578
\l_peek_token 150, 157, 175, 489, 630, 632, 636, 869, 951, 952, 953, 969, 981, 1003, 1022
\l_xeCJK_current_font_tl 1268, 1272, 2183, 2183, 2184, 2188, 2190
\l_xeCJK_family_tl 1736, 1748, 2184, 2191, 2218, 2221, 2222, 2224, 2228, 2231, 2261, 2268, 2269, 2277, 2277, 2278, 2371, 2377, 2398, 2506, 2514, 2525
\l_xeCJK_punct_style_tl 1268, 1270, 1576, 1588, 1635, 1639, 1653, 1657
\LA@space 3102
PlainEquation 3
SlantFactor 4
AllowBreakBetweenPuncts 5
\LocalConfig 2565
\LongPunct 1198

M

\makexeCJKactive 197, 201, 201
\makexeCJKinactive 198, 201, 202, 2821, 2822, 2824, 2828, 2838
EmboldenFactor 4
\mddefault 2453
\MiddlePunct 1198
\mode_if_vertical:T 3167
\msg_critical:nn 10
\msg_critical:nnn 20
\msg_error:nn 37
\msg_error:nnn 23
\msg_error:nnx 38
\msg_info:nnx 42
\msg_info:nnxx 43
\msg_new:nnn 3, 11, 36
\msg_redirect_module:nnn 2594, 2595, 2601, 2602
\msg_warning:nn 39
\msg_warning:nnx 40
\msg_warning:nnxx 41

N

indentfirst 3
\newCJKfontfamily 5, 2333, 2335
\NewDocumentCommand 201, 202, 320, 339, 389, 394, 401, 908, 1088, 1096, 1105, 1191, 1194, 1668, 1682, 1761, 2256, 2321, 2324, 2326, 2333, 2335,

2341, 2360, 2363, 2437, 2477, 2667, 2672, 2674, 2676, 2677, 2678, 2680, 2681, 2682, 2684, 2686, 2688, 2690, 2853
\NewLineCS 1042
\newXeTeXintercharclass 211
InlineEnv 4
InlineEnv+ 4
InlineEnv- 4
\nobreak 2639
\NoBreakCS 906
\NormalSpace 239
\normalspacedchars 9, 389, 389
EnvCS 3
EnvCS+ 3
EnvCS- 3

O

NoBreakCS 5
NoBreakCS+ 5
NoBreakCS- 5
LocalConfig 2
LongPunct 4
LongPunct+ 4
LongPunct- 4

P

\par 2637
\PassOptionsToPackage 2597, 2604, 2610, 2900, 2901
\pdfstringdefPreHook 2835
\pdftex_strcmp:D 128, 131
\peek_after:Nw 146, 152, 171, 176
\peek_catcode:NTF 946, 1013
\peek_meaning_remove:NTF 486
\penalty 2925, 2932
\Pifont 2827
\PlainEquation 1085
\prg_do_nothing: 414
\prg_new_conditional:Npnn 46, 106, 126, 188, 546, 1253, 1261, 2040, 2045, 2089
\prg_new_protected_conditional:Npnn 2249
\prg_return_false: 49, 109, 132, 192, 549, 1256, 1264, 2043, 2049, 2092, 2254
\prg_return_true: 49, 109, 129, 132, 192, 549, 1256, 1264, 2043, 2049, 2092, 2254
\ProcessKeysOptions 2646
\ProcessOptions 2902
\prop_clear:N 1917, 1959
\prop_get:Nn 1760
\prop_get:NnNF 2409
\prop_get:NnNT 1990, 2154, 2174
\prop_get:No 1756
\prop_get:NoNF 2397
\prop_get:NVNF 1772
\prop_get:NVNT 2137, 2143, 2376, 2449
\prop_get:NVNTF 2371
\prop_get:NxNF 2400
\prop_gpop:NnNT 1992
\prop_gput:Nnn 317, 2248
\prop_gput:NnV 2175
\prop_gput:Nvn 312
\prop_gput:NVV 2115, 2117, 2140, 2147
\prop_gput:Nxx 2240
\prop_if_empty:NTF 2412
\prop_map_function:NN 2017
\prop_map_inline:Nn 1885, 1918, 1962, 2096, 2104
\prop_new:N 33, 308, 1879, 2107, 2108, 2109
\prop_put:Nnn . 1880, 1881, 1882, 1883, 1884, 1971, 1973, 2001
\prop_put_if_new:Nnn 2003
\PunctStyle 1644
\punctstyle 2676, 2676
\PunctWidth 1198

Q

\q_stop 85, 86, 89, 186, 187
\quiet 2590

R

\r 2823, 2824
\RenewDocumentCommand 2722, 2827, 3114, 3130
\RenewDocumentEnvironment 3162
\RequirePackage 21, 24, 2647, 2649, 2865, 2872, 2875, 2887, 2905, 2906, 2907
\reverse_if:N 136
\rmdefault 2419, 2729

S

\scan_stop: 72, 123, 141, 673, 1039, 1419, 1441, 1451, 1518, 1541, 2536, 2886, 2902
\selectfont 2245, 2550, 2562
\seq_clear:N 1924
\seq_count:N 2780
\seq_gclear:c 1227
\seq_gput_right:cn 1231, 1241
\seq_gput_right:Nn 213, 224, 311, 1172, 1672, 1820, 2292
\seq_gput_right:Nv 214
\seq_gremove_all:cn 1250
\seq_gset_eq:NN 303
\seq_gset_from_clist:Nn 305
\seq_if_in:cnF 1238
\seq_if_in:NnF 890, 1065, 2290, 2784
\seq_map_function:NN 439, 2374
\seq_map_inline:cn 1225
\seq_map_inline:Nn 645, 903, 1081, 1147, 1163, 1610, 1925, 2113, 2123, 2792, 2974, 2976
\seq_new:c 1222
\seq_new:N 204, 205, 302, 304, 307, 879, 1077, 1087, 1675, 1817, 2297
\seq_put_right:Nn 890, 1066, 1832
\seq_remove_all:Nn 895, 1073
\seq_set_from_clist:Nn 1058
\seq_set_split:Nnn 884
\seq_use:Nnnn 1666
\setCJKfallbackfamilyfont 6, 7, 1761, 1761
\setCJKfamilyfont 5, 2308, 2333, 2333
\setCJKmainfont 5, 2304, 2321, 2321, 2323, 2328
\setCJKmathfont 6, 2329, 2437, 2437
\setCJKmonofont 5, 2306, 2321, 2326, 2331
\setCJKromanfont 2323, 2332
\setCJKsansfont 5, 2305, 2321, 2324, 2330
\SetSymbolFont 2454
\sfdefault 2420, 2730
\shapedefault 2453, 2455
\silent 2590
\skip_add:Nn 918
\skip_gset:Nn 2524
\skip_horizontal:N 923, 2492, 2926, 2933
\skip_horizontal:n 683, 2494, 2622, 3103, 3111

\skip_if_eq:nnTF 113
\skip_if_eq_p:nn 463, 521, 522
\skip_new:N 32, 574, 602, 2527
\skip_set:Nn 124, 2497, 3112
\skip_set_eq:NN 916, 3107
\skip_use:N 124
\skip_vertical:n 3146
\SlantedFeatures 1879
\SlantedFont 1879
\SlantFactor 1792
\SplitArgument 340
\str_case_x:nnn 1041, 2302, 2313, 2417, 2727
\str_case_x:non 1035
\str_if_eq:nnF 332, 479, 494, 2207
\str_if_eq:nnT 17, 2029
\str_if_eq:nnTF 1151, 1833, 1871, 2054, 2210, 2978
\str_if_eq_p:nn 1616, 1617, 2048, 2061, 2063
\str_if_eq_x:nnTF 81
\sw@slant 2741

T

\tex_baselineskip:D 2622
\tex_char:D 2886
\tex_font:D 104, 108, 115, 116, 117, 1605, 2518, 2519, 2543, 2551, 2558
\tex_fontdimen:D 115, 116, 117, 2518, 2543, 2551, 2558
\tex_global:D 2473
\tex_hbox:D 3169, 3182
\tex_hfil:D 3165
\tex_hss:D 3150
\tex_iftrue:D 2864
\tex_ignorespaces:D 72, 658, 662, 2275, 2344, 2366, 2670, 3127, 3139, 3170, 3183
\tex_indent:D 3167
\tex_italiccorrection:D 486, 487
\tex_kern:D 559, 560, 685, 3149
\tex_lastkern:D 548, 3110
\tex_lastskip:D 124, 463, 521, 522, 916, 918
\tex_let:D 2846
\tex_penalty:D 44, 45, 2925, 2932
\tex_rightskip:D 2497
\tex_romannumeral:D 154, 178
\tex_space:D 2924, 2931
\tex_spaceskip:D 113, 119
\tex_the:D 104, 412
\tex_unkern:D 626, 2748, 2754, 2760
\tex_unskip:D 466, 525, 919, 3173, 3180
\tex_vbox:D 3168
\tex_vrule:D 670
\tex_vss:D 3152
\textcentereddot 2818
\textellipsis 2818
\textemdash 2818
\textperiodcentered 2818
\textquotedblleft 2819
\textquotedblright 2819
\textquoteleft 2819
\textquoteright 2819
\textvisiblespace 2532
\tipaencoding 2822
\tl_case:Nnn 873
\tl_case:Non 869, 1022
\tl_clear:c 1920
\tl_clear:N 92, 902, 1080, 1774
\tl_concat:NNN 1045, 1051
\tl_const:Nn 111, 1196, 1197, 1661, 2439, 2894
\tl_const:Nx 51, 2451, 2648
\tl_gclear:N 93
\tl_gput_right:Nn 56, 58, 60, 2835
\tl_gput_right:Nx 313
\tl_gset:cn 1192, 1195
\tl_gset:cx 1278, 1280, 1282, 2513
\tl_gset:Nn 2573, 2662, 2663, 2664, 2665
\tl_gset:Nx 727, 737, 750, 767, 779, 788, 810, 819, 828, 837, 2415, 2578
\tl_gset_eq:cN 2353
\tl_if_blank:nTF 1348, 1825, 1964, 2051
\tl_if_blank:VT 1855
\tl_if_blank:VTF 1862
\tl_if_blank:vTF 2042
\tl_if_eq:NNTF 1576, 1635
\tl_if_exist:cF 1574, 1633, 2349, 2506
\tl_if_exist:cTF 48
\tl_if_exist:NF 2662, 2663, 2664, 2665
\tl_if_exist_p:c 1465
\tl_if_head_eq_meaning:nNTF 1876
\tl_if_head_eq_meaning:VNTF 1852
\tl_if_head_is_group_p:n 88
\tl_if_single_p:n 88
\tl_if_single_token_p:n 952
\tl_map_inline:nn 391, 889, 895, 1193, 2167, 2816
\tl_map_inline:xn 1192, 1228, 1236, 1247
\tl_new:c 1230, 1240, 2666
\tl_new:N 25, 26, 52, 53, 54, 316, 686, 878, 1032, 1084, 2183, 2277, 2582
\tl_put_left:Nn 2536, 2822
\tl_put_right:Nn 903, 1082, 1777, 1829, 2503
\tl_replace_all:Nnn 435, 1877
\tl_replace_all:NnV 1864
\tl_replace_once:Nnn 2740, 2741, 2945, 2947
\tl_set:cn 2848
\tl_set:Nn 142, 143, 169, 1771, 1873, 1940, 1942, 2184, 2278, 2882, 2889, 2890
\tl_set:Nx 83, 434, 1354, 1355, 1657, 1850, 2125, 2190, 2199, 2220, 2243, 2244, 2268, 2337, 2348, 2351, 2443, 2446, 2541, 2548, 2549, 2724
\tl_set_eq:cc 2849, 2850
\tl_set_eq:cN 1866
\tl_set_eq:NN 1653, 1857, 1861, 1863, 1872, 2269, 2402
\tl_to_str:n 2348
\tl_trim_spaces:n 90
\tl_use:c 1122, 1274, 1276, 1467
\token_get_arg_spec:N 951
\token_if_cs:NTF 969
\token_if_macro_p:N 630
\token_if_math_toggle_p:N 981
\token_if_other_p:N 632, 953
\token_if_space:NTF 489
\token_to_meaning:N 186
\token_to_str:N 1131, 1182, 2304, 2305, 2306, 2308, 2315, 2316, 2317, 2389, 2433, 2659, 2660
\TrimSpaces 320, 1089, 1668, 1682
\ttdefault 2421, 2731

U

\UL@hook 2908

\UL@leaders 3107, 3112
\UL@leadtype 2916, 2952, 2954
\UL@skip 3107, 3112
\UL@start 2995, 3008, 3019, 3027, 3036, 3053, 3073, 3087, 3097, 3104, 3121, 3126
\UL@stop 2995, 3008, 3016, 3023, 3032, 3041, 3067, 3070, 3084, 3093, 3104, 3116, 3123
\ULon 2945, 2946, 2947, 2948
PunctStyle *4*
PunctWidth *4*
\Unicode 2885, 2889, 2890
\use:c 319, 348, 548, 559, 560, 1434, 1435, 1437, 1447, 1448, 1529, 1530, 1532, 1795, 1803, 2237, 2456, 2525
\use:n 1049, 1368, 1372, 1399, 1409, 1500, 1533, 1539, 2737, 2755, 2761
\use:x 2510, 2856
\use_i:nn 1000, 1005, 1037, 2252
\use_ii:nn 314, 1082, 1534
\use_ii:nnn 1043
\use_iii:nnn 1024
\use_none:n . . 136, 137, 138, 350, 1365, 1369, 1538, 1980, 2763
\use_none:nn 1530, 2260
\use_none:nnnnn 2447
\usefont 2828
\UseInstance 1587, 1638
\usepackage 2659
AutoFakeBold *4, 5*
AutoFakeSlant *4, 5*
AutoFallBack *4*

V

\vbox_gset_to_ht:Nnn 3144
\verbatim@font 2503

X

\xCJKecglue 575
\xe@alloc@intercharclass 2780, 2782
\xeCJK@family 2269
\xeCJK@fix@penalty 2739, 2739, 2740, 2741
\xeCJK@italiccorr 2740, 2742, 2742
\xeCJK@setfont 2204
\xeCJK_add_font_features:n 2369, 2369, 2386
\xeCJK_add_font_features:x 2365
\xeCJK_allow_break: 44, 44, 1202
\xeCJK_app_inter_class_toks:nnn . . . 421, 421, 426, 480, 658
\xeCJK_app_inter_class_toks:nnx 2796
\xeCJK_Boundary_and_FullLeft:N 655, 743, 743
\xeCJK_Boundary_and_FullRight:N 657, 772, 772
\xeCJK_calc_punct_dimen:f 1583
\xeCJK_calc_punct_dimen:N 1593, 1593, 1625
\xeCJK_check_for_glue: 499, 505, 505, 2495
\xeCJK_check_FullRight: 846, 850, 856, 865, 865
\xeCJK_check_FullRight_symbol:Nw 851, 874, 874
\xeCJK_check_single:NNw 947, 955, 959, 959
\xeCJK_check_single:Nw 930, 933, 938, 944, 944
\xeCJK_check_single_cs:NNn 973, 974, 1020, 1020
\xeCJK_check_single_env:nnNn 1025, 1033, 1033
\xeCJK_check_single_equation:NNnNw . . 985, 986, 1011, 1011
\xeCJK_CJK_and_CJK:N 639, 640, 640, 930, 932, 933, 938, 939, 2973
\xeCJK_CJK_and_FullLeft:N 733, 733
\xeCJK_CJK_and_FullRight:N 784, 784
\xeCJK_class_group_begin: 442, 442, 448, 500, 723, 747, 763, 776, 3001, 3009
\xeCJK_class_group_end: . . . 442, 443, 454, 612, 702, 705, 716, 871, 2994, 3007, 3031, 3057, 3067, 3070, 3083, 3092
\xeCJK_class_num:n . . . 319, 319, 335, 385, 392, 409, 412, 1255
\xeCJK_clear_Boundary_and_CJK_toks: 438, 438, 451, 501, 726, 749, 766, 778, 3002, 3010
\xeCJK_clear_inter_class_toks:nn 413, 413, 441, 450, 725, 765
\xeCJK_copy_inter_class_toks:nnnn 427, 427, 1149, 1150, 1162, 1165, 1166, 2794, 2795, 2803, 2809
\xeCJK_cs_case_keys_define:nNNnn 876, 876, 906, 1042, 1048
\xeCJK_cs_clear:N 92, 92, 2838, 2918, 2919, 2952, 2955
\xeCJK_cs_gclear:N 92, 93
\xeCJK_declare_char_class:nn 325, 325, 342, 1142
\xeCJK_declare_char_class:nV . . 396, 397, 398, 399, 403, 404
\xeCJK_declare_char_class:nx 322, 1120
\xeCJK_declare_sub_char_class:nnn 1134, 1134, 1144
\xeCJK_declare_sub_char_class:nxx 1091
\xeCJK_def_node:nn 551, 551, 562, 563, 564, 565
\xeCJK_Default_and_FullLeft:nN 719, 719
\xeCJK_Default_and_FullRight:nN 759, 759
\xeCJK_fallback_loop:Nn 1736, 1738, 1738, 1745
\xeCJK_fallback_test_glyph:N . . 1718, 1721, 1726, 1732, 1732
\xeCJK_family_if_exist:x 2249
\xeCJK_family_if_exist:xF 2222
\xeCJK_family_if_exist:xT 1748, 2152, 2165
\xeCJK_family_if_exist:xTF 1740, 2249, 2266, 2284, 2442, 2445
\xeCJK_family_if_exist_use:x 2261, 2264, 2282, 2282
\xeCJK_font_gset_to_current:c 101, 101, 2192, 2201, 2232, 2544
\xeCJK_fontspec:nn 2346, 2346, 2358
\xeCJK_fontspec:VV 2382
\xeCJK_fontspec:xx 2343
\xeCJK_FullLeft_and_CJK: 642, 687, 687, 2970
\xeCJK_FullLeft_and_Default: 697, 697, 2969
\xeCJK_FullLeft_and_FullLeft:N 804, 804
\xeCJK_FullLeft_and_FullRight:N 813, 813
\xeCJK_FullRight_and_Boundary: 658, 660, 661, 661, 846, 848, 850, 856, 858
\xeCJK_FullRight_and_CJK: 644, 707, 707, 2972
\xeCJK_FullRight_and_Default: 662, 713, 713, 2971
\xeCJK_FullRight_and_FullLeft:N 823, 823
\xeCJK_FullRight_and_FullRight:N 832, 832
\xeCJK_FullRight_symbol:N 770, 782, 791, 821, 839, 849, 851, 859, 864, 864
\xeCJK_get_inter_class_toks:nn 411, 411, 418, 424, 430, 434, 2797, 2799, 2801, 2807
\xeCJK_get_punct_bounds:NN 721, 735, 745, 761, 774, 786, 807, 816, 825, 834, 867, 1572, 1572, 3065
\xeCJK_get_punct_kerning:NN 1631, 1631, 1643
\xeCJK_get_punct_kerning:oN 808, 817, 826, 835
\xeCJK_glue_to_skip:nN . . 121, 121, 571, 580, 597, 2922, 2929
\xeCJK_glyph_bounds:NN 1596, 1598, 1626, 1626
\xeCJK_glyph_if_exist:N 106
\xeCJK_glyph_if_exist:NTF 106, 1734, 1743
\xeCJK_glyph_if_exist_p:N 106
\xeCJK_gset_mathcode:nnnn 2463, 2468, 2468
\xeCJK_hook_for_ulem: 2908, 2909, 2909
\xeCJK_if_blank_x:n 126
\xeCJK_if_blank_x:nT 2800, 2806
\xeCJK_if_blank_x:nTF 126, 372, 1367, 2091, 2258
\xeCJK_if_blank_x_p:n 126, 370, 951
\xeCJK_if_last_node:n 546
\xeCJK_if_last_node:nT 476

\xeCJK_if_last_node:nTF 475, 546, 2746, 2752, 2758
\xeCJK_if_last_node_p:n 469, 470, 508, 513, 528, 530, 536, 537, 546
\xeCJK_if_package_loaded:n 46
\xeCJK_if_package_loaded:nF 2596, 2603, 2869
\xeCJK_if_package_loaded:nT 2710, 2825, 2833, 2874
\xeCJK_if_package_loaded:nTF 46, 61, 2608, 2700, 2871
\xeCJK_if_package_loaded_p:n 46, 2845
\xeCJK_if_same_class:NN 188
\xeCJK_if_same_class:NNTF 188, 1507
\xeCJK_if_same_class_p:NN 188
\xeCJK_ignore_spaces:w 614, 616, 616, 2496, 2770, 3160
\xeCJK_int_until_do:nn 135, 135, 351, 362, 2471
\xeCJK_inter_class_toks:nnc 651, 666
\xeCJK_inter_class_toks:nnn 408, 408, 410, 414, 446, 454, 458, 484, 497, 610, 639, 641, 643, 654, 656, 659, 2980, 2982, 2986
\xeCJK_inter_class_toks:nnx 417, 423, 429, 436, 649
\xeCJK_make_node:n 487, 490, 491, 551, 557, 613, 627, 2749, 2755, 2761, 3159
\xeCJK_new_class:n 206, 206, 239, 240, 241, 1138
\xeCJK_new_sub_key:n 1140, 1817, 1818, 1878
\xeCJK_no_break: 44, 45, 692, 695, 702, 705, 799, 800, 806, 815, 820, 838, 907, 921, 1015, 1016, 1027, 1028, 1204, 3048, 3051, 3067, 3072
\xeCJK_peek_after_ignore_spaces:nw 167, 167, 875
\xeCJK_peek_catcode_ignore_spaces:NTF . 139, 139, 618, 961
\xeCJK_pre_inter_class_toks:nnn 415, 415, 420, 495, 1153, 1167, 1169, 1176
\xeCJK_pre_inter_class_toks:nnx 2798
\xeCJK_punct_kerning_process:NN 1350, 1456, 1456
\xeCJK_punct_margin_process:NN 1349, 1352, 1352
\xeCJK_replace_inter_class_toks:nnnn 432, 432, 1157
\xeCJK_reverse:nnn 80, 80, 1358, 2088
\xeCJK_save_class:nn 217, 217, 234, 235, 236, 237, 238
\xeCJK_select_font: 449, 502, 724, 748, 764, 777, 1583, 2183, 2186, 2204, 2211, 2509, 3003
\xeCJK_select_font:x 1742, 2183, 2195
\xeCJK_set_char_class:nnn 334, 380, 380, 382, 2692
\xeCJK_set_family:nnn 1936, 1936, 1956
\xeCJK_set_family:Vnn 2354
\xeCJK_set_family:Vvv 1781, 2129
\xeCJK_set_family:xxx 1766, 2322, 2325, 2327, 2334, 2339, 2438
\xeCJK_set_family_fallback:nnN 1768, 1768, 1787
\xeCJK_set_family_fallback:xxN 1765
\xeCJK_set_mathfont: 2426, 2440, 2440
\xeCJK_set_visible_space_font: 2531, 2539, 2539
\xeCJK_swap_cs:NN 94, 94
\xeCJK_tl_remove_outer_braces:N 82, 82
\xeCJK_tl_remove_outer_braces:n 82, 83, 84, 1850
\xeCJK_token_value_charcode:N 184, 185, 1605, 1629
\xeCJK_token_value_class:N .. 183, 183, 191, 998, 1003, 1255
\xeCJK_visible_space_fallback: 2528, 2528, 2537
\xeCJKactive 194
\xeCJKallowbreakbetweenpuncts 2682, 2682
\xeCJKCancelSubCJKBlock 8, 1095, 1096
\xeCJKcaption 2851, 2853
\xeCJKDeclareCharClass 9, 320, 320
\xeCJKDeclarePunctStyle 9, 1668, 1668, 1681, 1689, 1690, 1691, 1698, 1706
\xeCJKDeclareSubCJKBlock 8, 1088, 1088, 1094, 1131
\xeCJKdisablefallback 2686, 2688
\xeCJKEditPunctStyle 9, 1682, 1682, 1688
\xeCJKenablefallback 2686, 2686
\xeCJKmainfont 2433
\xeCJKnobreak 11, 908, 908
\xeCJKnobreakbetweenpuncts 2682, 2684
\xeCJKplainchr 2676, 2677
\xeCJKResetCharClass 9, 401, 401, 407
\xeCJKResetPunctClass 9, 323, 394, 394, 405, 1092, 1102, 1111, 2693
\xeCJKRestoreSubCJKBlock 8, 1095, 1105
\xeCJKsetcharclass 2690, 2690
\xeCJKsetecglue 2679
\xeCJKsetemboldenfactor 2672, 2672
\xeCJKsetkern 9, 1194, 1194
\xeCJKsetslantfactor 2672, 2674
\xeCJKsetup .. 2, 1189, 2490, 2660, 2667, 2667, 2673, 2675, 2676, 2677, 2678, 2680, 2681, 2683, 2685, 2687, 2689, 2914
\xeCJKsetwidth 9, 1191, 1191
\xeCJKVerbAddon 11, 2477, 2477, 2502, 2503
\xetex_if_engine:F 10
\XeTeXcharclass 184, 354, 364, 387, 392
\XeTeXcharglyph 1629
\XeTeXdefaultencoding 2855, 2862
\XeTeXglyphbounds 1182, 1186, 1628
\XeTeXinterchartokenstate 201, 202, 2744
\XeTeXinterchartoks 409, 412
\XeTeXmathcode 2473