

xeCJK 宏包

CTEX.ORG

2026/07/22 v3.10.4*

目录

第 1 节 简介	1	3.7 其他	20
第 2 节 基本用法	2	第 4 节 已知问题和兼容性	20
第 3 节 用户手册	2	4.1 CJK 文字与命令交互时的间距	21
3.1 宏包选项	2	4.2 \char 原语与 interchar 的冲突	21
3.2 字体设置与选择	9	4.3 hyperref 链接与 CJKecglue	22
3.3 CJK 分区字体设置	13	第 5 节 xeCJK 代码实现	22
3.4 设置 CJK 字符范围	14	版本历史	222
3.5 标点符号的处理	14	代码索引	228
3.6 xeCJKfntef 用法说明	17		

第 1 节 简介

xeCJK 是一个 Xe_{La}TeX 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他空格, 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 Xe_{La}TeX 的一些最新特性, 需要 Xe_{La}TeX 0.9995.0 (2009/06/29) 以后的版本。xeCJK 依赖 L^AT_EX 3 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。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、李清²及黄晨成³。

*[ctex-kit rev. 257b9387](#).

¹leoliu.pku@gmail.com

²sobenlee@gmail.com

³liamhuang0205@gmail.com

第 2 节 基本用法

与其他 $\text{\LaTeX}$ 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:

例 1

```
\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 $\text{\XeTeX}$ 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。

第 3 节 用户手册

3.1 宏包选项

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

```
\xeCJKsetup \xeCJKsetup {\langle key_1 \rangle = \langle val_1 \rangle, \langle key_2 \rangle = \langle val_2 \rangle, \dots}
```

其中 `\langle key_1 \rangle`, `\langle key_2 \rangle` 是设置选项, 而 `\langle val_1 \rangle`, `\langle val_2 \rangle` 则是对应选项的设置内容。多个选项可以在一个语句中完成设置。例如

例 2

```
\usepackage[PunctStyle=kaiming]{xeCJK}
```

等价于

例 3

```
\usepackage{xeCJK}
.....
\xeCJKsetup{PunctStyle=kaiming}
```

带有 $\star$ 或者 $\star$ 标记的选项或命令只能在导言区中使用, 其中 $\star$ 还表示这个选项或命令只影响随后定义的 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, 则可以在导言区或正文中使用。**粗体**表示 `xeCJK` 的默认设置。

LocalConfig ★ `LocalConfig = {\true|false|name}`

是否使用本地配置文件 `xeCJK-⟨name⟩.cfg`。⟨name⟩ 可以是不包含空格的任意使文件名合法的字符串。如果设置为 `true`, 则使用的是 `xeCJK.cfg`; 设置为 `false` 则不载入配置文件。可以把将要在下文介绍到的对 **xeCJK** 的一些设置 (例如设置常用 CJK 字体、修改字符范围和定义新的标点输出格式等) 保存到文件 `xeCJK-⟨name⟩.cfg`。然后把这个文件放在本地的 TDS 目录下的适当位置。使用 $\TeX$ Live 的用户, 可以新建下列目录, 然后再把 `xeCJK-⟨name⟩.cfg` 放在里面:

`texlive/texmf-local/tex/xelatex/xecjk`

最后还需要在命令行下执行 `mktexlsr`, 刷新文件名数据库以便 $\TeX$ 系统能够找到它。

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

xeCJKactive `xeCJKactive = {\true|false}`

打开/关闭对中文的特殊处理。事实上, 这个选项会打开/关闭 $\XeTeX$ 的整个字符类机制, 依赖这个机制的宏包都会受到影响。

CJKspace `CJKspace = {\true|false}`

缺省状态下, **xeCJK** 会忽略 CJK 文字之间的空格, 使用这一选项来保留它们之间的空格。

CJKmath ★ `CJKmath = {\true|false}`

是否支持在数学环境中直接输入 CJK 字符。使用这个选项后, 可以直接在数学环境中输出 CJK 字符。`url` 宏包将一个 URL 放在一个特殊的数学环境中排版, 所以如果在 `\path` 等命令的路径参数中含有汉字, 则需要启用这个选项, 路径中的汉字才能显示。

CJKglue `CJKglue = {\hskip 0pt plus 0.08\baselineskip}`

设置 CJK 文字之间插入的 `glue`, 上边是 **xeCJK** 的默认值。一般来说, 除非有特殊需要 (例如, 改变文字间距等), 否则不需要设置这个选项, 使用默认值即可。如果要设置这个选项, 为了行末的对齐, 设置的 `glue` 最好有一定的弹性。

CJKecglue `CJKecglue = {\glue}`

设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距, 默认值是一个空格。使用这个选项设置的 `\glue` 最好也要有一定的弹性。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** 保留源文件中直接输入在 CJK 文字与西文或行内数学公式之间的空格; 这些空格使用普通词间空格。设为 `true` 后, **xeCJK** 会将它们替换为 `CJKecglue`。也可以直接把这个选项设置为一段 `\glue`; 这等价于先用该值设置 `CJKecglue`, 再设置 `xCJKecglue=true`。

注意: $\TeX$ 的节点列表只保存 `glue` 的数值, 不保存它来自源码空格还是显式 `\hskip`。在 `\mbox`、`\textcolor` 等命令或分组之后, 如果显式写出的 `glue` 与当前词间空格自然宽度相同, 并且带有 `shrink`, **xeCJK** 就无法判断它的来源, 可能按空格恢复规则将其替换为 `CJKecglue` 或 `CJKglue`。自然宽度不同或没有 `shrink` 的显式 `glue` 不受影响。如果必须原样保留前一种 `glue`, 请在它前面加零宽 `\kern0pt`, 也可以改用不同的自然宽度或去掉 `shrink`。这样检查会停在这枚 `kern`, 不再继续向前寻找 **xeCJK** 写入的 `marker`。

CheckSingle `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孤字的问题。如果这倒数三个字有作为控制序列的参数情况, 那么一般来说也不能正确处理。

WidowPenalty `WidowPenalty = {(penalty|10000)}`

使用 CheckSingle 选项后, 设置段末三个汉字之间的 penalty。初始值为 10 000, 即禁止在它们之间折行。

PlainEquation `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 = { \par \[ }`

NewLineCS+ 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。

NewLineCS-

EnvCS `EnvCS = { \begin \end }`

EnvCS+ 设置 $\text{\LaTeX}$ 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**

EnvCS-

InlineEnv `InlineEnv = {(env_1), (env_2), (env_3), ...}`

InlineEnv+ 在使用 CheckSingle 选项的时候, **xeCJK** 会将 CJK 文字后接着的 $\text{\LaTeX}$ 环境的开始 `\begin{...}`

InlineEnv- 和结束 `\end{...}` 视为断行的地方, 如果有某些特殊的 $\text{\LaTeX}$ 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。倾斜程度的取值范围是 $[-0.999, 0.999]$ 。

EmboldenFactor ☆ `EmboldenFactor = {(数字|4)}`

设置伪粗体的默认粗细程度。

SlantFactor ☆ `SlantFactor = {(数字|0.167)}`

设置伪斜体的倾斜程度, 范围是 $[-0.999, 0.999]$ 。

```
PunctStyle 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; 左标点后接右标点时保留字体的自然空白, 其余相邻标点
在字形边界允许时压缩到约 1.5 汉字宽度;

banjiao 半角式: 所有标点占半个汉字宽度;

kaiming 开明式: 句末点号用全角, 其他半角;

hangmobanjiao 行末半角式: 所有标点占一个汉字宽度, 行首行末对齐;

CCT CCT 格式: 所有标点符号的宽度略小于一个汉字宽度;

plain 原样(不调整标点间距)。

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

```
PunctFamily PunctFamily = {⟨false|family⟩}
```

默认情况下, CJK 标点符号的字体与 CJK 正文一致, `PunctFamily` 用于单独对标点符号设置字体。⟨*family*⟩ 需要使用随后说明的 `\setCJKfamilyfont` 或 `\newCJKfontfamily` 预先定义。`false` 表示取消本选项的作用, 让标点符号字体与正文一致。

```
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。同时属于 NoBreakLongPunct 的长标点则禁止在其前面断行。长标点与其他标点相
邻时, 断点两侧还须满足通常的标点禁则: 全角左标点不悬于行尾, 全角右标点不落于行首。
```

```
NoBreakLongPunct * NoBreakLongPunct = {⟨ …… ⟩}
```

```
NoBreakLongPunct+ *
NoBreakLongPunct- * 设置禁止在其前面断行的长标点。属于 NoBreakLongPunct 的字符必须同时属于 LongPunct。
与普通长标点不同, xeCJK 禁止在这类标点前面断行, 但仍允许在其后面断行, 并禁止在相同
长标点之间断行。缺省值包含省略号“…”和“…”。
```

```
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。
```

```
PoZheHaoLigature PoZheHaoLigature = {⟨true|false⟩}
```

New: 2026-07-04

一些设计优良的 **OpenType** 字体(如思源宋体、思源黑体)提供了破折号合字功能: 连续的两个 U+2014 会被替换为一个两倍字宽的字符。缺省状态下, **xeCJK** 在破折号之间插入的标点处理代码会阻断这一功能。启用这一选项后, **xeCJK** 把 U+2014 和 U+2015 归入 `PoZheHao` 字符类, 它们之间不插入任何内容, 从而恢复字体的合字功能。这一选项需要字体的配合(一般还需启用 `fwd` 或 `loc1` 等 **OpenType** 特性得到全角字形), 对不支持破折号合字的字体(多数中文字库)不要启用它, 否则连续破折号中间可能出现空隙。由于标点度量按字体缓存, 这一选项建议在导言区或使用相应字体排版破折号之前设置。例如:

例 4

```
etCJKmainfont{Noto Serif CJK SC}[RawFeature+=fwd]
eCJKsetup{PoZheHaoLigature}
```

LatinPunct LatinPunct = $\langle$ true|false $\rangle$

New: 2026-07-05 Unicode 中的弯引号 (U+2018、U+2019、U+201C、U+201D)、间隔号 (U+00B7) 与省略号 (U+2025、U+2026、U+2027) 等码位是中西文共用的。缺省状态下, **xeCJK** 把它们归入全角标点, 用中文字体输出全角字形; 在以西文为主的文档中, 夹在英文单词中间的撇号 (如 Children's 的 U+2019) 会因此变成突兀的全角形式。启用这一选项后, 这些共用标点改归入西文半角标点类 (HalfLeft / HalfRight), 使用当前西文字体输出, 也不再参与标点压缩。这一选项缺省未启用; LatinPunct 等价于 LatinPunct = true。破折号 (U+2014) 的归属不受此选项影响, 见 PoZheHaoLigature。例如:

例 5

```
eCJKsetup{LatinPunct} % 共用标点使用西文字体
```

这一选项可以在导言区或正文中随时切换, 作用范围是当前 $\text{T}_{\text{E}}\text{X}$ 分组。

CJLineBreak CJLineBreak = $\{\langle$ normal|strict $\rangle\}$

New: 2026-07-12 设置 Unicode Line_Break=CJ 的日文小假名等字符是否采用严格行首禁则。normal 将它们作为普通 CJK 文字处理, 保持历史默认行为; strict 保持同样的字距, 但在这些字符前插入 penalty 10 000, 防止它们出现在行首。例如きや和きやつ中的小假名前都不可断行。该选项可在分组内局部切换; 调用 \xeCJKResetCharClass 后仍保留当前设置。

PunctWidth ★ PunctWidth = $\{\langle$ length $\rangle\}$

缺省状态下, **xeCJK** 会根据所选择的标点处理格式自动计算标点所占的宽度, 如果对缺省设置不满意, 可以通过这一选项来改变它。为了使得标点所占的宽度能够适应字体大小的变化, 这里设置的 length 的单位最好用 em 等相对距离单位, 而不建议使用诸如 pt 之类的绝对距离单位。这里的设置可用于除了 plain 以外的所有标点处理格式。同时, 这里的设置对所有的 CJK 标点都生效, 如果只要设置部分标点, 请使用 3.5.1 节的 \xeCJKsetwidth。

PunctBoundWidth ★ PunctBoundWidth = $\{\langle$ length $\rangle\}$

与以上选项类似, 但设置的是标点符号出现在行首/尾时的宽度。

AllowBreakBetweenPuncts AllowBreakBetweenPuncts = $\langle$ true|false $\rangle$

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

RubberPunctSkip RubberPunctSkip = $\langle$ true|false|plus|minus $\rangle$

缺省状态下, 标点符号前/后的间距有一定的弹性。可以伸长到原始边界宽度, 可以收缩到标点另一侧的边界宽度。将本选项设置为 plus, 将只允许伸长; 设置为 minus 只允许收缩。设置为 false 将禁用这一特性, 从而使得前/后的间距为固定值。

CheckFullRight CheckFullRight = $\langle$ true|false $\rangle$

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

NoBreakCS NoBreakCS = { \footnote \footnotemark \nobreak }

NoBreakCS+ NoBreakCS- 设置不能在全角右标点后面断行的控制序列。以上是 **xeCJK** 的默认设置。如果这些控制序列在文档中只出现少量几次, 也可以不必使用 CheckFullRight 选项, 而是手工在这些控制序列前面加上 3.7 节介绍的 \xeCJKnobreak。

experiment/halfright-prebreakpenalty	experiment/halfright-prebreakpenalty = $\langle$ true false $\rangle$
New: 2026-05-03	这是一个实验性选项。缺省状态下, xeCJK 不会禁止在 CJK 文字或全角右标点后面紧跟的半角右标点(如. , ; : ! ?)] 等)之前换行。例如). 中, . 有可能被排在下一行的行首。打开此选项后, xeCJK 会在这类位置插入禁则(penalty 10 000), 防止半角右标点出现在行首。 此选项的效果类似于 LuaTeX-jā 中的 prebreakpenalty 机制, 但受限於 X_ƎTeX interchar 机制, 粒度为字符类级别而非逐字符级别。例如: <pre>\xeCJKsetup{experiment/halfright-prebreakpenalty}</pre>
experiment/punct-measure-fix	experiment/punct-measure-fix = $\langle$ true false $\rangle$
New: 2026-05-17	这是一个实验性选项。缺省状态下, L^AT_EX 的\para_end: 在执行\tex_par:D 之前会通过\unskip 移除水平列表末尾的 glue。如果段末恰好是全角标点, 其标点补偿 glue 也会被移除, 导致 tabulararray 等使用 \par 结束测量段落的宏包得到不正确的宽度。 打开此选项后, xeCJK 会在 para/end 钩子中插入一个 \kern, 补偿被 \unskip 移除的标点 glue 的自然宽度, 确保测量宽度与排版宽度一致。 注意: 此修复涉及一个权衡。 \kern 只保留了原 glue 的自然宽度, 丢弃了弹性分量 (stretch/shrink)。对于段末最后一行, \parfillskip 提供的 0pt plus 1fil 拉伸量会吸收所有剩余空间, 因此弹性丢失在绝大多数情况下无视觉影响。但在 CheckSingle 等强制 overfull 的极端场景下, 最后一行的 overfull 报告值可能增大。 使用示例: <pre>\xeCJKsetup{experiment/punct-measure-fix}</pre>

```
experiment/boundary-register experiment/boundary-register =
{
  command = <control sequence>,
  strategy = <strategy>, mode = <mode>
}
```

New: 2026-07-22

这是一个面向熟悉 $\text{\TeX}$ 节点和命令实现的用户实验性接口。它把 **xeCJK** 尚未处理的命令接入命令边界框架,使命令包装前后的可见内容尽量具有相同的中西文间距。一个键值只注册一个命令;需要注册多个命令时,可以重复使用本选项。

command 必须是一个控制序列。命令名含有 `@` 时,用户应在声明外使用 `\makeatletter` 和 `\makeatother`;注册公开的 $\text{\LaTeX}$ 命令时,应使用 `\ExplSyntaxOn` 和 `\ExplSyntaxOff` 让 `_` 和 `:` 成为命令名的一部分。这个接口不接受不带反斜线的命令名字字符串。

strategy 可以取以下值:

box 命令结束时在当前列表末尾留下一个水平盒子。

wrapped-box 命令会直接写出多个节点,并且能够在 **xeCJK** 打开的水平盒子中完整执行。

stream 命令的可见内容直接写入当前列表。

transparent 命令没有可见输出,只留下 PDF 锚点等特殊节点。

post-transparent 命令结束时留下零尺寸水平盒子,只能在命令执行后恢复边界状态。

下面的五个命令分别给出每种策略最简单的典型形式。实际命令只有在节点结构与相应示例一致时,才应采用同一种策略。

```
\newcommand\mybox[1]{\hbox{#1}}
\newcommand\mywrapped[1]{\kern0pt #1\kern0pt}
\newcommand\mystream[1]{\special{my-stream}#1}
\newcommand\myanchor{\special{my-anchor}}
\newcommand\myzero{\hbox{}}
\XeCJKsetup
{
  experiment/boundary-register = { command = \mybox,
    strategy = box },
  experiment/boundary-register = { command = \mywrapped,
    strategy = wrapped-box },
  experiment/boundary-register = { command = \mystream,
    strategy = stream },
  experiment/boundary-register = { command = \myanchor,
    strategy = transparent },
  experiment/boundary-register = { command = \myzero,
    strategy = post-transparent }
}
```

box、**wrapped-box** 和 **stream** 可以使用 **mode**。省略时默认为 **auto**,即采用运行时观察到的实际首、尾类别。**default** 把两端固定视为 **Default**; **first-default** 只把左端固定视为 **Default**,并且只适用于 **stream**。**transparent** 和 **post-transparent** 不接受 **mode**。

用户声明在导言区收集,到导言区末尾统一注册;即使声明写在分组内,也会对整篇文档生效。目标命令可以在普通的 `\AtBeginDocument` 中定义;如果它到正文开始以后才定义,注册不会生效并会给出错误。已经由 **xeCJK** 处理的命令和重复声明也会报错,本接口不提供反注册或覆盖内建注册的功能。命令 **hook** 安装完成后,如果其他代码重新定义了目标命令,新定义可能不再包含这里登记的处理。

并非所有命令都适合通用注册。`\verb`、`\url` 等使用特殊参数读取方式的命令可能需要专用适配;公开命令如果只负责读取参数,应当注册真正排版内容的内部命令。能够作为 **command** 读入的控制序列,也不一定能由 $\text{\LaTeX}$ 通用命令 **hook** 完整包住。通用注册不会分析命令参数;如果行内公式位于可见内容的开头或结尾, **auto** 未必能观察到公式边界。`\mbox` 等现有命令另有参数适配代码,用户注册不会自动取得这部分能力。

注册后应当把命令与“去掉命令包装后的相同可见内容”比较,并分别检查左右边界、四种源码空格以及 `\xCJKecglue` 的两个取值。随包提供的 **xeCJK-example-boundary-register.tex** 用明显不同的 `\CJKecglue` 和 `\CJKglue` 展示了这种检查方法。其中公式行还会显示上述参数适配限制;方框不等宽时,说明不能只靠所选通用策略处理该命令。

Verb Verb = $\langle \text{true} | \text{false} | \text{env} | \text{env+} \rangle$

true 表示在 `\verb` 命令或 `verbatim` 环境里不自动调整中英文之间的间距。env 选项在 `verbatim` 环境里自动计算中西文间距和中文之间的间距, 以便于保持代码的对齐; env 选项不调整 `\verb` 里的间距, env+ 选项还将正文里设置的间距应用到 `\verb` 里。这个选项对使用到 `\verbatim@font` 命令的情形均有效, 更一般的情况可以使用 3.7 节介绍的 `\xeCJKVerbAddon`。false 表示不作任何处理。以上选项的值除 false 外, 都禁止在汉字之间和汉字与西文之间自动换行。

LoadFandol ☆ LoadFandol = $\langle \text{true} | \text{false} \rangle$

当没有在导言区设置 CJK 字体时, 是否使用 Fandol 字体。如果启用这个选项, 需要安装 Fandol 字体系列。

3.2 字体设置与选择

\setCJKmainfont ☆ `\setCJKmainfont` $\langle \{ \langle \text{font name} \rangle \} [\langle \text{font features} \rangle]$ 或 `\setCJKmainfont` $[\langle \text{font features} \rangle] \{ \langle \text{font name} \rangle \}$

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

出于兼容性考虑, 字体属性可选项可以放在字体名称前面, 也可以放在后面。如果可选项放在后面, 字体名称与可选项之间不要有空格或者换行。

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

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

Mapping Mapping = $\langle \{ \langle \text{fullwidth-stop} | \text{full-stop} | \text{han-trad} | \text{han-simp} | \dots \rangle \}$

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

\setCJKsansfont ☆ `\setCJKsansfont` $\langle \{ \langle \text{font name} \rangle \} [\langle \text{font features} \rangle]$ 或 `\setCJKsansfont` $[\langle \text{font features} \rangle] \{ \langle \text{font name} \rangle \}$

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

\setCJKmonofont ☆ `\setCJKmonofont` $\langle \{ \langle \text{font name} \rangle \} [\langle \text{font features} \rangle]$ 或 `\setCJKmonofont` $[\langle \text{font features} \rangle] \{ \langle \text{font name} \rangle \}$

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

\setCJKfamilyfont ☆ `\setCJKfamilyfont` $\langle \{ \langle \text{family} \rangle \} \{ \langle \text{font name} \rangle \} [\langle \text{font features} \rangle]$ 或 `\setCJKfamilyfont` $\{ \langle \text{family} \rangle \} [\langle \text{font features} \rangle] \{ \langle \text{font name} \rangle \}$

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

`\CJKfamily` `\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 name>}[<font features>]` 或
`\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

例 6

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

等价于

例 7

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

`\CJKfontspec` `\CJKfontspec {<font name>}[<font features>]` 或
`\CJKfontspec [<font features>] {<font name>}`

在文档中定义新的 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

例 8

```
\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>}`
`\addCJKfontfeatures * {<font features>}`
`\addCJKfontfeatures [<block_1, block_2, ...>] {<font features>}`
`\addCJKfontfeatures * [<block_1, block_2, ...>] {<font features>}`

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

例 9

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

可以将文档中当前使用的 CJK 主分区字体放大为 1.1。

`\CJKrmdefault` 保存 `\textrm` 和 `\rmfamily` 所使用的 CJK 字体族, 默认值是 `rm`。

`\CJKsfdefault` 保存 `\textsf` 和 `\sffamily` 所使用的 CJK 字体族, 默认值是 `sf`。

`\CJKttdefault` 保存 `\texttt` 和 `\ttfamily` 所使用的 CJK 字体族, 默认值是 `tt`。

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

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

就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。

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

设置 CJK 字体族 `<family>` 的备用字体。例如, 使用

例 10

```
\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:

例 11

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

如果 `FallBack` 的值为空, 将设置的是备用字体。例如,

例 12

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

等价于

例 13

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

```
\setCJKfallbackfamilyfont * \setCJKfallbackfamilyfont {\family}
{
  {[font features_1]} {\font name_1}} ,
  {[font features_2]} {\font name_2}} ,
  .....
}[common font features] 或
\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

例 14

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

之后, 就设置了 SimSun 是 KaiTi_GB2312 的备用字体, 而 SimSun-ExtB 是 SimSun 的备用字体。若当前字体族缺字, 并没有备用字体, 则尝试使用 \CJKfamilydefault 的备用字体。

3.2.1 XeTeX 的字体名查找

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

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

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

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

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

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

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

例 15

```
\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⁴。这样列出的字体列表就比较简明易用, 如 Windows 下预装的中文字体:

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

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

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

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

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

3.3 CJK 分区字体设置

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

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

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

例 16

```
{ `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

例 17

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

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

例如, 可以使用

⁴由于汉字编码原因, Windows 下总需要把字体列表输出到文件中以防止乱码。

例 18

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

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

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

```
\xeCJKCancelSubCJKBlock {<block_1, block_2, ...>}
```

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

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

```
\xeCJKRestoreSubCJKBlock {<block_1, block_2, ...>}
```

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

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

3.4 设置 CJK 字符范围

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

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

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

用于恢复 `xeCJK` 对各个字符类别的初始化设置。

```
\xeCJKResetCharClass *
```

用于重置标点符号所属的字符类。

```
\xeCJKResetPunctClass *
```

```
\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 /、\、和 - (U+002D)。

3.5 标点符号的处理

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

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 plain 以外的所有标点处理格式。

```
\xeCJKsetwidth * {<标点列表>} {<length>}
```

```
\xeCJKsetwidth * {<标点列表>} {<length>}
```

`<标点列表>` 可以是单个标点，也可以是多个标点。例如，

例 19

```
\xeCJKsetwidth{。?}{0.7em}
```

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

```
\xeCJKsetkern * \xeCJKsetkern {<前标点>} {<后标点>} {<length>}
```

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

例 20

```
\xeCJKsetkern{: }{"}{0.3em}
```

将设置冒号与左双引号之间的空白宽度为 0.3em。

3.5.2 定义标点符号处理格式

```
\xeCJKDeclarePunctStyle * \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 * \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 *nan*。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

下面选项设置标点符号出现在行首或者行尾时的宽度或比例。

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

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

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

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

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

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

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

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

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`。

`abled-left-right-kerning` *<boolean>* 是否压缩左标点后紧跟右标点时二者之间的空白。默认值为 `true`; `quanjiao` 预设将它设为 `false`, 保留字体的自然空白。通过 `\xeCJKsetkern` 显式设置的字符对不受此选项影响。

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

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

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

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

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

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

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

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

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

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

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

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

例 21

```
\xeCJKDeclarePunctStyle { mine }
{
  fixed-punct-ratio      = nan ,
  fixed-margin-width     = 0 pt ,
  mixed-margin-width     = \maxdimen ,
  mixed-margin-ratio     = 0.5 ,
  middle-margin-width    = \maxdimen ,
  middle-margin-ratio    = 0.5 ,
```

```

add-min-bound-to-margin = true ,
bound-punct-width       = 0 em ,
enabled-hanging          = true ,
min-bound-to-kerning     = true ,
kerning-margin-minimum  = 0.1 em
}

```

就定义了一个名为 mine 的标点处理格式。可以在导言区通过

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

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

例 22

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

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

3.6 xeCJKfntef 用法说明

xeCJK 包含有一个子宏包 xeCJKfntef，可以用它来实现汉字加点和可断行的下划线等。它是 CJKfntef 宏包在 Xe_{La}TeX 下的替换版本，基本用法完全一致。

xeCJKfntef 基于 ulem 宏包，除了兼容 ulem 定义的一些命令外，还进行了一些扩充：

\CJKKunderline	\CJKKunderline [*] [-] [(选项)] {(内容)}
\CJKKunderdblline	
\CJKKunderwave	虚室生白, 吉祥止止
\CJKKsout	虚室生白, 吉祥止止
\CJKKxout	虚室生白, 吉祥止止

虚室生白, 吉祥止止
 虚室生白, 吉祥止止
 虚室生白, 吉祥止止
 虚室生白, 吉祥止止
 虚室生白, 吉祥止止

```

1 \CJKKunderline{虚室生白, 吉祥止止}\\
2 \CJKKunderdblline{虚室生白, 吉祥止止}\\
3 \CJKKunderwave{虚室生白, 吉祥止止}\\
4 \CJKKsout{虚室生白, 吉祥止止}\\
5 \CJKKxout{虚室生白, 吉祥止止}

```

```

1 \CJKKunderline-{南朝}\CJKKunderline-{梁}\CJKKunderline-{劉勰}%
2 \CJKKunderwave-{文心雕龍}\CJKKunderwave-{養氣}\\
3 \CJKKunderline*[thickness=1pt, hidden=true]{瞻彼閼者, 虚室生白, 吉祥止止}

```

南朝梁劉勰文心雕龍養氣

\CJKKunderdot	\CJKKunderdot [(选项)] {(内容)}
---------------	-----------------------------

在汉字下加点点，可以和上述下划线命令嵌套使用。例如

虚室生白, 吉祥止止
 虚室生白, 吉祥止止

```

1 \CJKKunderline{虚室生白, \CJKKunderdot{吉祥}止止}\\
2 \CJKKunderdot{虚室生白, \CJKKunderline{吉祥}止止}

```

对上述六种对象，xeCJKfntef 提供了一些选项，设置点或线的位置和颜色。可以用 \xeCJKsetup 预先统一设置它们，也可以在使用时特别设置。

```

skip \xeCJKsetup { underline/skip = <true|false> }
      \xeCJKsetup { underline = { skip = <true|false> }, ... } }

```

默认情况下，下划线会自动跳过中文标点符号，可以设置本选项为 false，禁用这一功能。相应下划线命令后加上 * 号，具有相同的效果。

subtract 设置本选项为 true, 使得下划线的首尾减少一定距离, 避免前后的下划线连在一起, 适用于古籍标点整理中的专名号和书名号。在相应下划线命令后加上 - 号, 具有相同的效果。

hidden 设置本选项为 true, 将隐藏文本内容, 只画下划线。

format `\xeCJKsetup { underline/format = \color{red} }`
`\xeCJKsetup { underwave = { format = \color{red}, ... } }`
 设置线或点的格式, 比如颜色。

textformat 设置下划线或点的正文的格式。例如:

```
1 \CJKUnderline[textformat=\color{blue}]{虚室生白, 吉祥止止}\
2 \CJKUnderdot[textformat=\bfseries, format=\color{red}]{虚室生白, 吉祥止止}
```

虚室生白, 吉祥止止

虚室生白, 吉祥止止

symbol 设置 \CJKUnderwave 或 \CJKUnderdot 的符号。

例如, 波浪线 \CJKUnderwave 的符号不会随字号而变化, 在小字号下不好看。我们可以将它改为随字号而变化大小:

瞻彼阒者, 虚室生白, 吉祥止止

```
1 % \usepackage{fix-cm}
2 \xeCJKsetup{
3   underwave/symbol=
4     \fontsize{0.5em}{0pt}%
5     \fontencoding{U}\fontfamily{lasy}\selectfont
6     \char 58\relax
7 \footnotesize
8 \CJKUnderwave{瞻彼阒者, 虚室生白, 吉祥止止}
```

thickness 设置 \CJKUnderline、\CJKUnderdblline 和 \CJKsout 的线的厚度。初始值是 \ULthickness。

depth 设置线或点的深度(基线到线或点的顶部的距离)。初始值都是 0.2em。

boxdepth \CJKUnderdot 可能会影响到行距, 可以设置本选项进行调整。如果不希望 \CJKUnderdot 影响行距, 可以将本选项设置为 0pt。

sep 设置 \CJKUnderdot 与 \CJKUnderline、\CJKUnderdblline 或 \CJKUnderwave 嵌套使用时, 点与线或者线与点的距离。

gap 设置 \CJKUnderdblline 的两条线之间的距离。初始值是 1.1pt。

height 设置删除线 \CJKsout 的高度(线的中心到基线的距离)。初始值是 0.35em。

例如, 我们可以设置 \CJKsout 的厚度和颜色, 让它具有类似高亮的效果:

```
1 \CJKsout*[thickness=2.5ex, format=\color{yellow}]{瞻彼阒者, 虚室生白, 吉祥止止}
```

瞻彼阒者, 虚室生白, 吉祥止止

xeCJKfntef 还供给了自定义下划线和符号的 \CJKUnderanyline 和 \CJKUnderanyline。

\CJKUnderanyline `\CJKUnderanyline [*] [-] [<选项>] {<深度>} {<下划内容>} {<文本内容>}`

xeCJKfntef 先将 <下划内容> 放进一个盒子(\xeCJKfntefbox)里, 然后向下移动 <深度> 给定的距离, 再用于填充。可用的 <选项> 是 textformat、skip、hidden、subtract、sep 和 boxdepth。选项 sep 和 boxdepth 的初始值是空, 表示禁用该选项的功能。可以在 \xeCJKsetup 中通过对象 ulem 来设置。

例如, 高亮效果也可以如下实现:

1 \CJKunderanyline*{0.5ex}{\color{yellow}\rule{2pt}{2.5ex}}{虛室生白，吉祥止止}

虛室生白，吉祥止止。

`\CJKunderansymbol \CJKunderansymbol` [`选项`] `{(深度)}` `{(符号)}` `{(文本内容)}`

`\xeCJKIntef` 将 ⟨符号⟩ 放进一个盒子 (`\xeCJKfntefbox`) 里。⟨深度⟩ 参数用于设置盒子顶部的深度 (基线到盒子顶部的距离)。可用的 ⟨选项⟩ 是 `textformat`、`sep` 和 `boxdepth`, 意义与 `\CJKunderdot` 的相同。

例如,给汉字加三角形,可以如下设置:

1 \CJKunderansymbol[sep=0.1em]{0.2em}{\tiny\$\triangle\$}

2 {瞻彼閼者，虚室生白，\CJKunderline{吉祥止止}}

瞻彼閼者，虚室生白，吉祥止止。

`\xeCJKfontefon` `\xeCJKfontefon` `[*]` `[-]` `[<选项>]`

功能与用法 `ulem` 宏包的 `\ULon` 相同，扩展了可选参数符号 `*` 和 `-`，可用的〈选项〉是 `textformat`、`skip`、`hidden` 和 `subtract`。这四个选项对 `ulem` 宏包定义的 `\uline` 等命令也有效，需要在 `\xeCJKsetup` 中通过对象 `ulem` 来设置。例如

```
1 \xeCJKsetup{ulem={textformat=\bfseries\color{red}, skip=true}}
2 \uline{虛室生白，吉祥止止}
```

虛室生白，吉祥止止。

xeCJKfntef 的下划线和删除线命令内部使用 **ulem** 宏包的 \hbox 机制排版文字。汉字 \CJKsout{文} 中 \CJKsout 前后的源码空格会被 xeCJK 自动处理为 CJK 文字间距 (CJKglue)，与汉字 文字 汉字的效果一致。关于 xeCJKfntef 命令与颜色命令的嵌套组合等更多细节，请参见“已知问题和兼容性”一节中关于“CJK 文字与命令交互时的间距”的说明。

此外，`xeCJKfontef` 还提供了指定宽度，让汉字分散对齐的环境 `CJKfilltwsides` 和 `CJKfilltwsides*`。

CJKfilltwosides	\begin{CJKfilltwosides} [⟨位置⟩] {⟨宽度⟩}
	文本内容\\
	文本内容
	\end{CJKfilltwosides}

环境中的内容被放入垂直盒子中, 可选参数 `<位置>` 指定盒子的基线位置。可以使用 `t` (顶部)、`c` (居中) 和 `b` (底部), 默认是 `c`。`<宽度>` 参数指定盒子的宽度。`CJKfillltwosides*` 环境与 `CJKfillltwosides` 的区别是, 当 `<宽度>` 不大于零或者不大于盒子的自然宽度时, 就取盒子的自然宽度。例如

瞻 彼 閼 者，
虛室生白，吉祥止止。

```

1 \begin{CJKfilltwosides}{.8\linewidth}
2 瞻彼閼者，\
3 虚室生白，吉祥止止
4 \end{CJKfilltwosides}

```

瞻彼閼者，
虛室生白，吉祥止止。

```

1 \begin{CJKfillltwosides*}{Opt}
2 瞻彼閼者，\\
3 虚室生白，吉祥止止
4 \end{CJKfillltwosides*}

```

3.7 其他

`\xeCJKVerbAddon`
`\xeCJKOffVerbAddon`

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

`\xeCJKnobreak` ……汉字。`\xeCJKnobreak\footnote{脚注}`

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

`\xeCJKShipoutHook`

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

第 4 节 已知问题和兼容性

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

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

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

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

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

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

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

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

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

来取消 `\cprotect` 的外部宏限制。但由于 `\cprotect` 的特殊性, 应该只在外部使用它, 即不要让它出现在任何宏的参数中。其他 `\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 $\text{\XeTeX}$ 的 `\XeTeXinterchartoks` 机制, 与使用相同机制的宏包 (例如 `polyglossia` 和 `xesearch`) 可能会存在不同程度的冲突。`xeCJK` 虽然为此作了一些处理, 但与它们共同使用时应该小心。

4.1 CJK 文字与命令交互时的间距

`xeCJK` 利用 $\text{\XeTeX}$ 的 `interchar` 机制, 在 CJK 文字之间自动插入 `CJKglue`, 在 CJK 文字与西文之间自动插入 `CJKecglue`, 并自动忽略 CJK 文字间的源码空格。当 CJK 文字被 $\text{\LaTeX}$ 命令包裹时, 命令引入的分组、盒子或特殊节点可能干扰这一机制。`xeCJK` 的预期行为是:

前 中 后	% 直接排版
前 <code>\foo{中}</code> 后	% 用命令包裹

两种写法应产生相同的 CJK 间距。

目前 `xeCJK` 已确保以下命令在此场景下的间距行为正确:

- 字体切换命令: `\textbf`、`\textit`、`\emph` 等。这些命令仅切换字体, 不改变正文的边界类别。
- 颜色命令: `\textcolor`、`\colorbox` 等 (包括 $\text{\LaTeX} 2_{\epsilon}$ 和 $\text{\LaTeX} 3$ 的颜色后端)。
- `xeCJKfntef` 命令: `\CJKunderdot`、`\CJKunderline`、`\CJKsout`、`\CJKunderwave`、`\CJKunderdblline`、`\CJKxout`。
- 盒子命令: `\mbox`、`\fbox`、`\makebox`、`\framebox` 等。
- `hyperref` 的链接注释: `\ref`、`\hyperref` 等。
- 以上命令的嵌套组合, 例如:

前 <code>\CJKunderline{\textcolor{red}{中}}</code> 后
前 <code>\textcolor{red}{\CJKunderline{中}}</code> 后

行内公式也按相同原则处理。`$x$`、`\(x\)` 和 `\ensuremath{x}` 的边界与直接写出的行内公式一致; 上述命令包裹公式时, 不应把公式误当成普通西文。公式旁有源码空格时, `xeCJKecglue=false` 保留普通词间空格, `xeCJKecglue=true` 则将其替换为 `CJKecglue`。

上述命令共用一套边界记录和间距恢复机制, 但每类命令仍须提供明确的开始和结束位置, 不能自动识别任意宏的内部节点。未列出的第三方命令如果在 CJK 文字间引入特殊的分组或节点结构, 仍可能产生多余或缺失的间距。如果遇到间距异常, 欢迎到 [GitHub Issues](#) 提交兼容请求。

4.2 \char 原语与 interchar 的冲突

$\text{\XeTeX}$ 的 `interchar` 机制工作在 token 层面, 无法区分 `\char` 原语的输出和直接字符输入。因此当第三方宏包在内部使用 `\char` 从非 CJK 字体取字形时, 如果该字符编码恰好属于 `xeCJK` 的 `interchar` 字符分类, 就会被拦截并切换到 CJK 字体, 产生错误的字形。例如 `mtpro2` 的 `\overbrace` 通过 `\char183` 取花括号字形, 而 `U+00B7` (`MIDDLE DOT`) 在 `xeCJK` 中属于 `FullRight` 类, 导致花括号变成中文间隔号。

为此, `xeCJK` 提供 `\xeCJKchar` 命令, 用法与 `\char` 完全相同, 但在输出字符前临时关闭 `\XeTeXinterchartokenstate`。用户可以使用 `\xeCJKchar"00B7` 代替 `\char"00B7` 来避免 `interchar` 拦截。

`xeCJK` 已为以下宏包的命令提供了自动兼容补丁:

- `mtpro2`: `\overbrace`、`\underbrace`

- `pifont:\Pifont` (U 编码字体切换)

如果遇到其他宏包的类似问题, 可以在导言区手动包装有冲突的命令:

```
% 方法一: 直接用 \xeCJKchar 替换 \char
\hbox{\myfont\xeCJKchar183}

% 方法二: 包装第三方命令, 关闭 interchar
\let\origProblemCmd\problemCmd
\renewcommand\problemCmd[1]{%
  \makexeCJKinactive % 关闭 interchar
  \origProblemCmd{#1}}
```

`\makexeCJKinactive` 是赋值操作, 受 $\TeX$ 分组约束——在 $\{\dots\}$ 、 $\$...\$$ 或 $\begin{group}...\end{group}$ 内使用时, 结束后自动恢复。

4.3 hyperref 链接与 CJKecglue

`hyperref` 的链接注释命令 (如 `\ref`、`\hyperref` 等) 在内部会产生 `pdf:bann / pdf:eann` 特殊节点。`xeCJK` 对此做了兼容处理, 确保链接注释不影响正常的 `CJKecglue` 插入。例如, 以下写法的排版效果与不加 `\ref` 时等价:

```
图\ref{fig}图 % 等价于"图1图"
```

无论是否加载 `hyperref`, `\ref` 前后都会自动插入适当的 `CJKecglue`。

第 5 节 xeCJK 代码实现

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

5.1 运行环境检查

`xeCJK` 必须使用 $\text{Xe}\TeX$ 引擎的支持。

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

应该使用较新版本的 `expl3` 宏包。

```
11 \msg_new:nnn { xeCJK } { l3-too-old }
12 {
13   Support~package~`#1'~too~old. \\
14   Please~update~an~up~to~date~version~of~the~bundles\\
15   `l3kernel'~and~`l3packages'\\
16   using~your~TeX~package~manager~or~from~CTAN.\\
17   \str_if_eq:nnT {#1} { expl3 } { Loading~xeCJK~will~abort! }
18 }
19 \ifpackagelater { expl3 } { 2025/10/09 } { }
20 { \msg_critical:nnn { xeCJK } { l3-too-old } { expl3 } }
```

`\ctex_disable_package:n` 由 `ctexhook` 提供, `\ctex_patch_cmd:Nnn` 由 `ctexpatch` 提供。

```
21 \RequirePackage { ctexhook }
22 \RequirePackage { ctexpatch }
```

`\xeCJK_if_package_loaded:n` 判断宏包是否被引入, 可用于文档正文中。

```
\xeCJK_if_package_loaded:nTF
23 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
24 {
25   \tl_if_exist:cTF { ver@ #1 . \c__xeCJK_package_ext_tl }
26   { \prg_return_true: } { \prg_return_false: }
27 }
28 \tl_const:Nc \c__xeCJK_package_ext_tl { \@pkgextension }
```

下面这些 CJK 系列宏包不应该被使用。

```
29 \msg_new:nnn { xeCJK } { after-package }
30 {
31   The~`#1'~package~and~xeCJK~are~incompatible.\\\
32   Please~load~it~after~xeCJK.
33 }
34 \clist_map_inline:nn { CJKnumb }
35 {
36   \xeCJK_if_package_loaded:nT {#1}
37   { \msg_error:nnn { xeCJK } { after-package } {#1} }
38 }
39 \clist_map_inline:nn
40 { CJKulem , CJKvert , CJKpunct , CJKutf8 , CJK }
41 { \ctex_disable_package:n {#1} }
```

将 CJKfntef 包替换为 xeCJKfntef 包。

```
42 \ctex_if_format_at_least:nTF { 2020/10/01 }
43 { \ctex_replace_package:nn { CJKfntef } { xeCJKfntef } }
44 { \ctex_disable_package:n { CJKfntef } }

45 \cs_if_exist:Nf \NewDocumentCommand
46 { \RequirePackage { xparse } }
47 \RequirePackage { xtemplate }
```

5.2 内部工具

分配临时变量。

```
48 \tl_new:N \l__xeCJK_tmp_tl
49 \tl_new:N \l__xeCJK_tmptb_tl
50 \int_new:N \l__xeCJK_tmp_int
51 \box_new:N \l__xeCJK_tmp_box
52 \dim_new:N \l__xeCJK_tmp_dim
53 \bool_new:N \l__xeCJK_tmp_bool
54 \skip_new:N \l__xeCJK_tmp_skip
55 \clist_new:N \l__xeCJK_tmp_clist
```

`\__xeCJK_msg_new:nn` 各种信息函数的缩略形式。

```
\__xeCJK_error:n
\__xeCJK_error:ne
\__xeCJK_warning:ne
\__xeCJK_info:nee

56 \cs_new_protected:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
57 \cs_new_protected:Npn \__xeCJK_msg_new:nnn { \msg_new:nnnn { xeCJK } }
58 \cs_new_protected:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
59 \cs_new_protected:Npn \__xeCJK_error:ne { \msg_error:nne { xeCJK } }
60 \cs_new_protected:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
61 \cs_new_protected:Npn \__xeCJK_warning:ne { \msg_warning:nne { xeCJK } }
62 \cs_new_protected:Npn \__xeCJK_warning:nee { \msg_warning:nnee { xeCJK } }
63 \cs_new_protected:Npn \__xeCJK_warning:neee { \msg_warning:nneee { xeCJK } }
64 \cs_new_protected:Npn \__xeCJK_info:nee { \msg_info:nnee { xeCJK } }

\xeCJK_allow_break:
\xeCJK_no_break:
65 \cs_new_protected:Npn \xeCJK_allow_break:
66 { \tex_penalty:D \c_zero_int }
67 \cs_new_protected:Npn \xeCJK_no_break:
68 { \tex_penalty:D \c__xeCJK_nobreak_penalty_int }
69 \int_const:Nn \c__xeCJK_nobreak_penalty_int { 10 000 }
```

在 \document 前后和宏包后加上各种钩子, 依赖 ctexhook。

```

\__xeCJK_at_end_preamble:n
\__xeCJK_after_preamble:n
\__xeCJK_after_end_preamble:n
\__xeCJK_package_hook:nn
70 \AtBeginDocument { \xeCJK@document@hook }
71 \ctex_at_end_preamble:n { \xeCJK@document@left@hook }
72 \ctex_after_end_preamble:n { \xeCJK@document@right@hook }
73 \cs_new_protected:Npn \xeCJK@document@hook
74 { \tl_use:N \g__xeCJK_after_preamble_hook_tl }
75 \cs_new_protected:Npn \xeCJK@document@left@hook
76 { \tl_use:N \g__xeCJK_at_end_preamble_hook_tl }
77 \cs_new_protected:Npn \xeCJK@document@right@hook
78 { \tl_use:N \g__xeCJK_after_end_preamble_hook_tl }
79 \cs_new_protected:Npn \__xeCJK_at_end_preamble:n
80 { \tl_gput_right:Nn \g__xeCJK_at_end_preamble_hook_tl }
81 \cs_new_protected:Npn \__xeCJK_after_preamble:n
82 { \tl_gput_right:Nn \g__xeCJK_after_preamble_hook_tl }
83 \cs_new_protected:Npn \__xeCJK_after_end_preamble:n
84 { \tl_gput_right:Nn \g__xeCJK_after_end_preamble_hook_tl }
85 \cs_new_protected:Npn \__xeCJK_package_hook:nn
86 { \ctex_at_end_package:nn }
87 \tl_new:N \g__xeCJK_at_end_preamble_hook_tl
88 \tl_new:N \g__xeCJK_after_preamble_hook_tl
89 \tl_new:N \g__xeCJK_after_end_preamble_hook_tl

```

\xeCJKShipoutHook 在 \shipout 盒子里加钩子, 可以影响到页眉页脚。 \AtBeginDvi 将参数保存在盒子中, 而 atbegshi 的 \AtBeginShipout 在 \shipout 盒子构建好之后才起作用, 所以它们都影响不到页眉页脚。我们通过往 \@begindvi 里加入钩子来完成。注意, 第一次使用 \@begindvi 之后, 它会将自身定义为 \@empty。

```

90 \__xeCJK_after_preamble:n
91 { \tl_put_right:Nn \@begindvi { \xeCJK@first@begindvi } }
92 \cs_new_protected:Npn \xeCJK@first@begindvi
93 {
94   \xeCJKShipoutHook
95   \cs_if_exist:NTF \@begindvi
96     { \tl_gput_right:Nn }
97     { \tl_const:Nn }
98   \@begindvi { \xeCJKShipoutHook }
99 }
100 \NewDocumentCommand \xeCJKShipoutHook { }
101 {
102   \bool_if:NF \l__xeCJK_shipout_hook_bool
103   {
104     \bool_set_true:N \l__xeCJK_shipout_hook_bool
105     \tl_use:N \l__xeCJK_shipout_hook_tl
106   }
107 }

```

\xeCJK_add_to_shipout:n 往 \shipout 盒子中加入钩子。

```

108 \cs_new_protected:Npn \xeCJK_add_to_shipout:n
109 { \tl_put_right:Nn \l__xeCJK_shipout_hook_tl }
110 \tl_new:N \l__xeCJK_shipout_hook_tl
111 \bool_new:N \l__xeCJK_shipout_hook_bool

```

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

```

112 \cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
113 {
114   \exp_last_unbraced:Ne
115   \__xeCJK_tl_remove_outer_braces:w { \tl_trim_spaces:n {#1} } \s_stop
116 }
117 \cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \s_stop
118 {
119   \tl_if_single:nTF {#1}
120   {
121     \tl_if_head_is_N_type:nTF {#1}
122     { \tl_trim_spaces:n }

```

```

123      { \xeCJK_tl_remove_outer_braces:n }
124    }
125    { \tl_trim_spaces:n }
126    {#1}
127  }

```

\xeCJK_cs_clear:N 让控制序列的意义为空。

```

\xeCJK_cs_gclear:N
128 \cs_new_protected:Npn \xeCJK_cs_clear:N #1
129 { \cs_set_eq:NN #1 \prg_do_nothing: }
130 \cs_new_protected:Npn \xeCJK_cs_gclear:N #1
131 { \cs_gset_eq:NN #1 \prg_do_nothing: }

```

\xeCJK_swap_cs:NN 交换 #1 和 #2 的意义。

```

132 \cs_new_protected:Npn \xeCJK_swap_cs:NN #1#2
133 {
134   \cs_set_eq:NN \__xeCJK_swap_cs_aux:w #1
135   \cs_set_eq:NN #1 #2
136   \cs_set_eq:NN #2 \__xeCJK_swap_cs_aux:w
137   \cs_undefine:N \__xeCJK_swap_cs_aux:w
138 }

```

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

```

139 \cs_new_protected:Npn \xeCJK_font_gset_to_current:N
140 { \exp_after:wN \__xeCJK_font_gset_to_current_aux:NN \tex_the:D \tex_font:D }
141 \cs_new_protected:Npn \__xeCJK_font_gset_to_current_aux:NN #1#2
142 { \cs_if_eq:NNF #1 \tex_nullfont:D { \cs_gset_eq:NN #2#1 } }

```

\xeCJK_glyph_if_exist_p:N 判断当前字体中是否含有字符 #1。fontspec 中的类似函数在判断为真的时候, 会留有一个
 \xeCJK_glyph_if_exist:NTF \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。

```

143 \prg_new_conditional:Npnn \xeCJK_glyph_if_exist:N #1 { p , T , F , TF }
144 {
145   \tex_iffontchar:D \tex_font:D `#1 \exp_stop_f:
146   \prg_return_true: \else: \prg_return_false: \fi:
147 }

```

\c_xeCJK_space_skip_tl 当前字体状态下, 一个字间空格产生的 glue 的长度, 包括伸展和收缩部分。

```

148 \tl_const:Nn \c_xeCJK_space_skip_tl
149 {
150   \int_compare:nNnTF \g__xeCJK_space_factor_int = { 1000 }
151   {
152     \skip_if_eq:nNnTF \tex_spaceskip:D \c_zero_skip
153     {
154       \tex_fontdimen:D 2 ~ \tex_font:D
155       plus \tex_fontdimen:D 3 ~ \tex_font:D
156       minus \tex_fontdimen:D 4 ~ \tex_font:D
157     }
158     { \tex_spaceskip:D }
159   }
160   {
161     \skip_if_eq:nNnTF \tex_spaceskip:D \c_zero_skip
162     {
163       \int_compare:nNnTF \g__xeCJK_space_factor_int < { 2000 }
164       {
165         \__xeCJK_space_skip_scale:nnn
166         { \tex_fontdimen:D 2 ~ \tex_font:D }
167       }
168       {
169         \skip_if_eq:nNnTF \tex_xspaceskip:D \c_zero_skip
170         {
171           \__xeCJK_space_skip_scale:nnn
172           {
173             \tex_fontdimen:D 2 ~ \tex_font:D +
174             \tex_fontdimen:D 7 ~ \tex_font:D
175           }

```

```

176         }
177         { \tex_xspaceskip:D \use_none:nn }
178     }
179     { \tex_fontdimen:D 3 ~ \tex_font:D }
180     { \tex_fontdimen:D 4 ~ \tex_font:D }
181 }
182 {
183     \int_compare:nNnTF \g__xeCJK_space_factor_int < { 2000 }
184     { \__xeCJK_space_skip_scale:nnn { \tex_spaceskip:D } }
185     {
186         \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
187         {
188             \__xeCJK_space_skip_scale:nnn
189             {
190                 \tex_spaceskip:D +
191                 \tex_fontdimen:D 7 ~ \tex_font:D
192             }
193         }
194         { \tex_xspaceskip:D \use_none:nn }
195     }
196     { \tex_gluestretch:D \tex_spaceskip:D }
197     { \tex_glueshrink:D \tex_spaceskip:D }
198 }
199 }
200 }
201 \cs_new:Npn \__xeCJK_space_skip_scale:nnn #1#2#3
202 {
203     \dim_eval:n {#1}
204     plus \fp_eval:n { \g__xeCJK_space_factor_int / 1000 } #2
205     minus
206     \int_div_truncate:nn
207     { 1000 * \int_value:w #3 } { \g__xeCJK_space_factor_int } sp
208 }

```

`\xeCJK_reset_space_factor:` 在 `\XeTeXinterchartoks` 里, `\spacefactor` 已经被重置为 1000。我们需要在 `Default` 类里保存 `\spacefactor` 用于计算空格宽度。

```

209 \int_new:N \g__xeCJK_space_factor_int
210 \cs_new_protected:Npn \xeCJK_reset_space_factor:
211 { \int_gset:Nn \g__xeCJK_space_factor_int { 1000 } }
212 \xeCJK_reset_space_factor:

```

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

```

213 \cs_new_protected:Npn \xeCJK_glue_to_skip:nN #1#2
214 {
215     \group_begin:
216     \hbox_set:Nw \l__xeCJK_tmp_box #1 \scan_stop:
217     \__xeCJK_if_last_glue:TF
218     {
219         \exp_args:NNNo \hbox_set_end:
220         \skip_set:Nn #2 { \skip_use:N \tex_lastskip:D }
221     }
222     {
223         \exp_args:NNNo \hbox_set_end:
224         \skip_set:Nn #2 { \dim_use:N \box_wd:N \l__xeCJK_tmp_box }
225     }
226     \exp_args:NNNo \group_end:
227     \skip_set:Nn #2 { \skip_use:N #2 }
228 }

```

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

```

229 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
230 {
231     \__xeCJK_int_until_do:wn \use_none:n
232     { \reverse_if:N \if_int_compare:w #1#2 }
233 }

```

```

234 \cs_new_protected:Npn \__xeCJK_int_until_do:wn \use_none:n #1
235 { #1 \exp_after:wN \__xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
236 \int_new:N \l__xeCJK_begin_int
237 \int_new:N \l__xeCJK_end_int

```

`\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。

```

238 \cs_new_protected:Npn \xeCJK_peek_catcode_ignore_spaces:NTF #1#2#3
239 {
240   \cs_set_eq:NN \l__xeCJK_peek_search_token #1 \scan_stop:
241   \cs_set_protected:Npe \__xeCJK_peek_catcode_true:w
242   { \exp_not:N \group_align_safe_end: \exp_not:n {#2} }
243   \cs_set_protected:Npe \__xeCJK_peek_catcode_false:w
244   { \exp_not:N \group_align_safe_end: \exp_not:n {#3} }
245   \bool_set_false:N \l__xeCJK_peek_ignore_spaces_bool
246   \group_align_safe_begin:
247   \peek_after:Nw \__xeCJK_peek_catcode_ignore_spaces_branches:w
248 }
249 \cs_new_protected:Npn \__xeCJK_peek_catcode_ignore_spaces_branches:w
250 {
251   \if_meaning:w \l_peek_token \c_space_token
252     \bool_set_true:N \l__xeCJK_peek_ignore_spaces_bool
253     \exp_after:wN \peek_after:Nw
254     \exp_after:wN \__xeCJK_peek_catcode_ignore_spaces_branches:w
255     \tex_romannumeral:D 0
256   \else:
257     \if_catcode:w
258       \exp_not:N \l_peek_token \exp_not:N \l__xeCJK_peek_search_token
259       \exp_after:wN \exp_after:wN
260       \exp_after:wN \__xeCJK_peek_catcode_true:w
261     \else:
262       \exp_after:wN \exp_after:wN
263       \exp_after:wN \__xeCJK_peek_catcode_false:w
264     \fi:
265   \fi:
266 }
267 \cs_new_eq:NN \l__xeCJK_peek_search_token ?
268 \cs_new_eq:NN \__xeCJK_peek_catcode_true:w \prg_do_nothing:
269 \cs_new_eq:NN \__xeCJK_peek_catcode_false:w \prg_do_nothing:
270 \bool_new:N \l__xeCJK_peek_ignore_spaces_bool

```

`\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。

```

271 \cs_new:Npn \xeCJK_token_value_class:N #1
272 { \tex_XeTeXcharclass:D \xeCJK_token_value_charcode:N #1 }

```

`\xeCJK_token_value_charcode:N` 当记号 #1 的 `charcode` 大于等于 0x10000 时, XeTeX 0.9999.0 版以前的 `\meaning` 的返回结果比较特殊⁵, 需要特别处理。0.9999.0 版以后的 XeTeX 的 `\meaning` 对于超出 BMP 的字符, 会返回两个字符, 分别对应于其 UTF-16 编码的首尾代理⁶。这一 Bug 在 TeX Live 2015 的 0.99992 版中得到修复⁷。

```

273 \cs_new:Npn \xeCJK_token_value_charcode:N #1
274 { \exp_after:wN \__xeCJK_get_charcode:w \token_to_meaning:N #1 \q_stop }
275 \group_begin:
276   \cs_set:Npn \__xeCJK_tmp:w #1 ~ #2 ~ #3#4#5 \q_stop
277   {
278     \tl_if_empty:nTF { #4#5 }
279     {
280       \cs_new:Npn \__xeCJK_get_charcode:w ##1 ~ ##2 ~ ##3 \q_stop
281       { \int_eval:n { `##3 } }
282     }

```

⁵ 参见 <http://tug.org/pipermail/xetex/2013-January/023967.html> 和 <http://tex.stackexchange.com/a/64848>。

⁶ 参见 <http://tug.org/pipermail/xetex/2013-June/024543.html>。

⁷ 参见 <http://tug.org/pipermail/xetex/2015-May/025941.html>

```

283     {
284         \tl_if_empty:nTF {#5}
285         {
286             \cs_new:Npn \__xeCJK_get_charcode:w ##1 ~ ##2 ~ ##3##4 \q_stop
287             {
288                 \int_eval:n
289                 {
290                     \tl_if_empty:nTF { ##4 }
291                     { `##3 }
292                     { ( `##3 - "D800 ) * "400 + ( `##4 - "DC00 ) + "10000 }
293                 }
294             }
295         }
296         {
297             \cs_new:Npn \__xeCJK_get_charcode:w ##1 ~ ##2 ~ ##3##4 \q_stop
298             { \int_eval:n { \tl_if_empty:nTF { ##4 } { `##3 } { "20000 } } }
299         }
300     }
301 }
302 \exp_after:wN \__xeCJK_tmp:w \token_to_meaning:N ~~~~~20000 { } \q_stop
303 \group_end:

```

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

```

\xeCJK_if_CJK_class:N__
304 \prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
305 {
306     \if_cs_exist:w
307     \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 }
308     \cs_end:
309     \prg_return_true: \else: \prg_return_false: \fi:
310 }
311 \cs_new:Npn \__xeCJK_CJK_class_tl:n #1
312 { c__xeCJK_CJK_class_ \int_eval:n {#1} _tl }

```

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

```

\xeCJK_if_same_class:NN__
313 \prg_new_conditional:Npnn \xeCJK_if_same_class:NN #1#2 { p , T , F , TF }
314 {
315     \if_int_compare:w \xeCJK_token_value_class:N #1 =
316     \xeCJK_token_value_class:N #2 \exp_stop_f:
317     \prg_return_true: \else: \prg_return_false: \fi:
318 }

```

\xeCJK_make_boundary: 利用 \scan_stop: 结束 CJK 分组, 用于恢复字体等。

```

319 \cs_new_protected:Npn \xeCJK_make_boundary:
320 { \bool_if:NT \l__xeCJK_CJK_group_bool { \scan_stop: } }

```

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。

```

321 \keys_define:nn { xeCJK / options }
322 {
323     xeCJKActive .choice: ,
324     xeCJKActive / true .code:n = { \makexeCJKActive } ,
325     xeCJKActive / false .code:n = { \makexeCJKInactive } ,
326     xeCJKActive .default:n = { true }
327 }

\makexeCJKActive 328 \NewDocumentCommand \makexeCJKActive { }
\makexeCJKInactive 329 { \tex_XeTeXinterchartokenstate:D = \c_one_int }
330 \NewDocumentCommand \makexeCJKInactive { }
331 { \tex_XeTeXinterchartokenstate:D = \c_zero_int }

```

抑制 BOM 和零宽格式字符, 避免触发错误的 inter-class toks 插入。

```
332 \char_set_catcode_ignore:n { "FEFF }
333 \char_set_catcode_ignore:n { "200B }
334 \char_set_catcode_ignore:n { "200C }
335 \char_set_catcode_ignore:n { "200D }
336 \char_set_catcode_ignore:n { "2060 }
```

5.4 字符类别设定

分别用于记录在 xeCJK 中使用的字符类别名称和新建的字符类别的编号。

```
\g__xeCJK_class_seq
\g__xeCJK_new_class_seq
```

```
337 \seq_new:N \g__xeCJK_class_seq
338 \seq_new:N \g__xeCJK_new_class_seq
```

新建一个字符类别。#1 是自定义名称。

```
\xeCJK_new_class:n
```

```
339 \cs_new_protected:Npn \xeCJK_new_class:n #1
340 {
341   \int_if_exist:cTF { \__xeCJK_class_csname:n {#1} }
342   { \__xeCJK_error:ne { class-already-defined } {#1} }
343   {
344     \exp_args:Nc \newXeTeXintercharclass
345     { \__xeCJK_class_csname:n {#1} }
346     \clist_new:c { g__xeCJK_#1_range_clist }
347     \seq_gput_right:Nn \g__xeCJK_class_seq {#1}
348     \seq_gput_right:Nv \g__xeCJK_new_class_seq
349     { \__xeCJK_class_csname:n {#1} }
350   }
351 }
```

保存 XeTeX 预定义的字符类别。#1 是自定义名称, #2 是编号。

```
\xeCJK_save_class:nn
```

```
352 \cs_new_protected:Npn \xeCJK_save_class:nn #1#2
353 {
354   \int_if_exist:cTF { \__xeCJK_class_csname:n {#1} }
355   { \__xeCJK_error:ne { class-already-defined } {#1} }
356   {
357     \int_const:cn { \__xeCJK_class_csname:n {#1} } {#2}
358     \clist_new:c { g__xeCJK_#1_range_clist }
359     \seq_gput_right:Nn \g__xeCJK_class_seq {#1}
360   }
361 }
```

字符类名称对应的控制序列名字。

```
\__xeCJK_class_csname:n
```

```
362 \cs_new:Npn \__xeCJK_class_csname:n #1 { c__xeCJK_#1_class_int }
363 \cs_new_eq:cN { \__xeCJK_class_csname:n { Others } } \l__xeCJK_tmp_int
364 \__xeCJK_msg_new:nn { class-already-defined }
365 {
366   XeTeX~character~class~`#1'~has~been~already~defined.\\
367   Please~take~another~name. \\
368 }
```

xeCJK 需要以下字符类别用于字符输出。其中 Default、CJK、FullLeft、FullRight、Boundary 为 XeTeX 中预定义的类别, xeCJK 新增加了 HalfLeft、HalfRight、NormalSpace 和 CM。其中异体字选择符 (Ideographic Variation Selectors)⁸ 需要 XeTeX 0.9999.0 以上的版本⁹和相关字体的支持。

⁸<http://www.unicode.org/reports/tr37/>

⁹<http://tug.org/pipermail/xetex/2013-March/024118.html>

类别	说明	例子
Default	西文一般符号	abc123
CJK	CJK 表意符号	汉字 あいう
FullLeft	全角左标点	(《: “
FullRight	全角右标点	, 。)》”
HalfLeft	半角左标点	([{
HalfRight	半角右标点	,.?)]}
NormalSpace	前后原始间距的符号	/
Boundary	边界	空格
CM	组合标识	异体字选择符
HangulJamo	朝鲜文字母（仅兼容旧代码，不再分配字符）	/
HangulJamoL	朝鲜文字母初声	ㄴ
HangulJamoV	朝鲜文字母中声	ㅣ
HangulJamoT	朝鲜文字母终声	ㅏ
CJStarter	日文条件行首禁则字符	やつ
PoZheHao	支持合字的破折号	——

```

Default 369 \xeCJK_save_class:nn { Default } { 0 }
CJK
FullLeft XeTeX 0.99994 将字符类总数扩大到 409610。
FullRight
Boundary 370 \str_const:Nn \c__xeCJK_xetex_version_str
371 { \int_use:N \tex_XeTeXversion:D \tex_XeTeXrevision:D }
372 \fp_compare:nNnTF { \c__xeCJK_xetex_version_str } > { 0.99993 }
373 { \xeCJK_save_class:nn { Boundary } { 4095 } }
374 { \xeCJK_save_class:nn { Boundary } { 255 } }

```

LaTeX 2_ε 2016/02/01 不再预设置 CJK 字符类。

```

375 \int_compare:nNnTF { \tex_XeTeXcharclass:D "4E00 } = \c_one_int
376 {
377   \xeCJK_save_class:nn { CJK } { 1 }
378   \xeCJK_save_class:nn { FullLeft } { 2 }
379   \xeCJK_save_class:nn { FullRight } { 3 }
380   \int_const:Nn \c__xeCJK_class_begin_int { 3 }
381 }
382 {
383   \xeCJK_new_class:n { CJK }
384   \xeCJK_new_class:n { FullLeft }
385   \xeCJK_new_class:n { FullRight }
386   \int_const:Nn \c__xeCJK_class_begin_int { 0 }
387 }

```

HalfLeft 新增西文半角左/右标点、前后原始间距的符号和异体字选择符类，以及支持破折号合字的字
HalfRight 符类。
NormalSpace

```

CM 388 \xeCJK_new_class:n { HalfLeft }
HangulJamo 389 \xeCJK_new_class:n { HalfRight }
HangulJamoL 390 \xeCJK_new_class:n { NormalSpace }
HangulJamoV 391 \xeCJK_new_class:n { CM }
HangulJamoT 392 % 旧的 |HangulJamo| 类只为兼容用户代码保留；新代码应按音节位置使用
CJStarter 393 % |HangulJamoL|、|HangulJamoV| 和 |HangulJamoT|。
PoZheHao 394 \xeCJK_new_class:n { HangulJamo }
395 \xeCJK_new_class:n { HangulJamoL }
396 \xeCJK_new_class:n { HangulJamoV }
397 \xeCJK_new_class:n { HangulJamoT }
398 \xeCJK_new_class:n { CJStarter }
399 \xeCJK_new_class:n { PoZheHao }

```

¹⁰<http://tug.org/pipermail/xetex/2016-February/026363.html>

西文半角左/右标点和前后原始间距的字符类。

```
\c__xeCJK_HalfLeft_chars_clist
\c__xeCJK_HalfRight_chars_clist
\c__xeCJK_NormalSpace_chars_clist
400 \clist_const:Nn \c__xeCJK_HalfLeft_chars_clist
401 { "28 , "5B , "60 , "7B , "2329 }
402 \clist_const:Nn \c__xeCJK_HalfRight_chars_clist
403 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , "232A }
404 \clist_const:Nn \c__xeCJK_NormalSpace_chars_clist { "2D , "2F , "5C }
```

以下对全角标点符号的归类来源于 X_YTeX 的脚本 `unicode-char-prep.pl` 和 Unicode 数据库¹¹。

\c__xeCJK_OP_chars_clist Open Punctuation (OP)

U+2018	‘	U+201C	“	U+3008	〈	U+300A	《	U+300C	「	U+300E	『	U+3010	【
U+3014	〔	U+3016	〔	U+3018	〔	U+301A	〔	U+301D	〰	U+FE17	ㄟ	U+FE35	（
U+FE37	ㄿ	U+FE39	ㄿ	U+FE3B	ㄿ	U+FE3D	ㄿ	U+FE3F	ㄿ	U+FE41	ㄿ	U+FE43	ㄿ
U+FE47	ㄿ	U+FE59	（	U+FE5B	{	U+FE5D	{	U+FF08	（	U+FF3B	〔	U+FF5B	{
U+FF5F	（	U+FF62	〔										

以下代码的第一行是中西文共用的左引号。

```
405 \clist_const:Nn \c__xeCJK_OP_chars_clist
406 {
407   "2018 , "201C ,
408   "3008 , "300A , "300C , "300E , "3010 , "3014 , "3016 , "3018 , "301A , "301D ,
409   "FE17 , "FE35 , "FE37 , "FE39 , "FE3B , "FE3D , "FE3F , "FE41 , "FE43 , "FE47 ,
410   "FE59 , "FE5B , "FE5D , "FF08 , "FF3B , "FF5B , "FF5F , "FF62
411 }
```

\c__xeCJK_PR_chars_clist Prefix Numeric (PR)

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

```
412 \clist_const:Nn \c__xeCJK_PR_chars_clist
413 { "FE69 , "FF04 , "FFE1 , "FFE5 , "FFE6 }
```

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

```
414 \clist_const:Nn \c__xeCJK_FullLeft_chars_clist
415 {
416   \exp_not:V \c__xeCJK_OP_chars_clist ,
417   \exp_not:V \c__xeCJK_PR_chars_clist
418 }
```

\c__xeCJK_CL_chars_clist Close Punctuation (CL)

U+00B7	·	U+2019	,	U+201D	”	U+2013	—	U+2014	—	U+2025	⋯	U+2026	⋯
U+2027	·	U+2E3A	⋯	U+3001	、	U+3002	。〃	U+3009	〉	U+300B	》	U+300D	」
U+300F	』	U+3011	】	U+3015	〕	U+3017	〃	U+3019	〕	U+301B	〕	U+301E	」
U+301F	ㄿ	U+FE11	、	U+FE12	。〃	U+FE18	ㄿ	U+FE36	ㄿ	U+FE38	ㄿ	U+FE3A	ㄿ
U+FE3C	ㄿ	U+FE3E	ㄿ	U+FE40	ㄿ	U+FE42	ㄿ	U+FE44	ㄿ	U+FE48	ㄿ	U+FE50	,
U+FE52	.	U+FE5A	)	U+FE5C	}	U+FE5E	)	U+FF09	)	U+FF0C	,	U+FF0E	.
U+FF3D	]	U+FF5D	}	U+FF60	)	U+FF61	。	U+FF63	」	U+FF64	、		

以下代码的第一行是中西文共用的一些标点符号。

```
419 \clist_const:Nn \c__xeCJK_CL_chars_clist
420 {
421   "00B7 , "2019 , "201D , "2013 , "2014 , "2025 , "2026 , "2027 , "2E3A ,
422   "3001 , "3002 , "3009 , "300B , "300D , "300F , "3011 , "3015 , "3017 , "3019 ,
423   "301B , "301E , "301F , "FE11 , "FE12 , "FE18 , "FE36 , "FE38 , "FE3A , "FE3C ,
424   "FE3E , "FE40 , "FE42 , "FE44 , "FE48 , "FE50 , "FE52 , "FE5A , "FE5C , "FE5E ,
425   "FF09 , "FF0C , "FF0E , "FF3D , "FF5D , "FF60 , "FF61 , "FF63 , "FF64
426 }
```

¹¹<http://www.unicode.org/reports/tr14/>

\c__xeCJK_NS_chars_clist Nonstarter (NS)

U+30FB	・	U+FE54	；	U+FE55	：	U+FF1A	：	U+FF1B	；	U+FF65	・	U+16FE0	☒
--------	---	--------	---	--------	---	--------	---	--------	---	--------	---	---------	---

Hyphens (cl-03)

U+301C	～	U+30A0	=	U+FF5E	～
--------	---	--------	---	--------	---

Iteration marks (cl-09)

U+3005	々	U+303B	ゝ	U+309D	ゝ	U+309E	ゞ	U+30FD	ゝ	U+30FE	ゞ
--------	---	--------	---	--------	---	--------	---	--------	---	--------	---

根据 W3C 的资料¹², cl-03 和 cl-09 在非常松散的情况下可以没有禁则。我们将全角浪线 U+FF5E 等连接号归入 FullRight 类并在宏包末尾设为 LongPunct。

```

427 \clist_const:Nn \c__xeCJK_hyphens_chars_clist
428 { "301C , "30A0 , "FF5E }
429 \clist_const:Nn \c__xeCJK_iteration_marks_chars_clist
430 { "3005 , "303B , "309D , "309E , "30FD , "30FE }
431 \clist_const:Nn \c__xeCJK_NS_chars_clist
432 { "30FB , "FE54 , "FE55 , "FF1A , "FF1B , "FF65 , "16FE0 }
433 \AtEndOfPackage
434 {
435   \cs_set:Npn \__xeCJK_tmp:w #1
436     { \char_generate:nn {#1} { 12 } }
437   __xeCJK_add_special_punct:nn { long }
438     { \clist_map_function:NN \c__xeCJK_hyphens_chars_clist \__xeCJK_tmp:w }
439   \cs_undefine:N \__xeCJK_tmp:w
440 }

```

\c__xeCJK_EX_chars_clist Exclamation/Interrogation (EX)

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

```

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

```

\c__xeCJK_IS_chars_clist Infix Numeric Separator (IS)

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

```

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

```

\c__xeCJK_CJ_chars_clist Conditional Japanese Starter (CJ)。这类字符的禁则是可选的¹³, 为实现的简单计, 我们把它们归入 CJK 类, 即没有禁则。

U+3041	あ	U+3043	い	U+3045	う	U+3047	え	U+3049	お	U+3063
U+3085	ゆ	U+3087	よ	U+308E	わ	U+3095	か	U+3096	け	U+30A1
U+30A5	ウ	U+30A7	エ	U+30A9	オ	U+30C3	ツ	U+30E3	ヤ	U+30E5
U+30EE	ワ	U+30F5	カ	U+30F6	ケ	U+30FC	一	U+31F0	ク	U+31F1
U+31F3	ト	U+31F4	ヌ	U+31F5	ハ	U+31F6	ヒ	U+31F7	フ	U+31F8
U+31FA	ム	U+31FB	ラ	U+31FC	リ	U+31FD	ル	U+31FE	レ	U+31FF
U+FF68	イ	U+FF69	ウ	U+FF6A	エ	U+FF6B	オ	U+FF6C	ヤ	U+FF6D
U+FF6F	ツ	U+FF70	一	U+1B12	☒	U+1B150->"1B152	☒->"1B152	U+1B155	☒	U+1B164->"1B167

```

444 \clist_const:Nn \c__xeCJK_CJ_chars_clist
445 {
446   "3041 , "3043 , "3045 , "3047 , "3049 , "3063 , "3083 , "3085 , "3087 , "308E ,
447   "3095 , "3096 , "30A1 , "30A3 , "30A5 , "30A7 , "30A9 , "30C3 , "30E3 , "30E5 ,
448   "30E7 , "30EE , "30F5 , "30F6 , "30FC , "31F0 , "31F1 , "31F2 , "31F3 , "31F4 ,
449   "31F5 , "31F6 , "31F7 , "31F8 , "31F9 , "31FA , "31FB , "31FC , "31FD , "31FE ,
450   "31FF , "FF67 , "FF68 , "FF69 , "FF6A , "FF6B , "FF6C , "FF6D , "FF6E , "FF6F ,
451   "FF70 , "1B12 , "1B150 -> "1B152 , "1B155 , "1B164 -> "1B167
452 }

```

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

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

```

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

```

¹²<http://www.w3.org/TR/jlreq/>

¹³<https://github.com/CTeX-org/ctex-kit/issues/165>

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

```

454 \clist_const:Nc \c__xeCJK_FullRight_chars_clist
455 {
456   \exp_not:V \c__xeCJK_CL_chars_clist ,
457   \exp_not:V \c__xeCJK_NS_chars_clist ,
458   \exp_not:V \c__xeCJK_EX_chars_clist ,
459   \exp_not:V \c__xeCJK_IS_chars_clist ,
460   \exp_not:V \c__xeCJK_PO_chars_clist ,
461   \exp_not:V \c__xeCJK_hyphens_chars_clist
462 }
```

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

```

463 \clist_const:Nc \c__xeCJK_CJK_chars_clist
464 {
```

- 闽南语、客家话阴去和阳去声调标记

```

465   "02EA -> "02EB ,
```

- CJK Radicals Supplement (中日韩部首补充)

```

466   "2E80 -> "2EFF ,
```

- Kangxi Radicals (康熙字典部首)

```

467   "2F00 -> "2FDF ,
```

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

```

468   "2FF0 -> "2FFF ,
```

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

```

469   "3000 -> "303F ,
```

- Hiragana (日文平假名)

```

470   "3040 -> "309F ,
```

- Katakana (日文片假名)

```

471   "30A0 -> "30FF ,
```

- Bopomofo (注音字母)

```

472   "3100 -> "312F ,
```

- Hangul Compatibility Jamo (谚文兼容字母)

```

473   "3130 -> "318F ,
```

- Kanbun (象形字注释标志)

```

474   "3190 -> "319F ,
```

- Bopomofo Extended (注音字母扩展)

```

475   "31A0 -> "31BF ,
```

- CJK Strokes (中日韩笔画)

```

476   "31C0 -> "31EF ,
```

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

```

477   "31F0 -> "31FF ,
```

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

```

478   "3200 -> "32FF ,
```

- CJK Compatibility (中日韩兼容)

```

479   "3300 -> "33FF ,
```

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

480 "3400 -> "4DBF ,

- Yijing Hexagrams Symbols (易经六十四卦符号)

481 "4DC0 -> "4DFF ,

- CJK Unified Ideographs (中日韩统一表意文字)

482 "4E00 -> "9FFF ,

- Yi Syllables (彝文音节)

483 "A000 -> "A48F ,

- Yi Radicals (彝文字根)

484 "A490 -> "A4CF ,

- Hangul Syllables (谚文音节)

485 "AC00 -> "D7AF ,

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

486 "F900 -> "FAFF ,

- Vertical Forms (竖排形式)

487 "FE10 -> "FE1F ,

- CJK Compatibility Forms (中日韩兼容形式)

488 "FE30 -> "FE4F ,

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

489 "FF00 -> "FFEF ,

- Ideographic Symbols and Punctuation (表意文字符号及标点)

490 "16FE0 -> "16FFF ,

- Tangut (西夏文)

491 "17000 -> "187FF ,

- Tangut Components (西夏文部首)

492 "18800 -> "18AFF ,

- Khitan Small Script (契丹小字)

493 "18B00 -> "18CFF ,

- Tangut Supplement (西夏文增补)

494 "18D00 -> "18D7F ,

- Tangut Components Supplement (西夏文部首增补)

495 "18D80 -> "18DFF ,

- Kana Extended-B (日文假名扩展 B)

496 "1AFF0 -> "1AFFF ,

- Kana Supplement (日文假名增补)

497 "1B000 -> "1B0FF ,

- Kana Extended-A (日文假名扩展 A)

498 "1B100 -> "1B12F ,

- Small Kana Extension (小型日文假名扩展)

499 "1B130 -> "1B16F ,

- Nushu (女书)

```
500      "1B170 -> "1B2FF ,
```

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

```
501      "1F200 -> "1F2FF ,
```

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

```
502      "20000 -> "2A6DF ,
```

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

```
503      "2A700 -> "2B73F ,
```

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

```
504      "2B740 -> "2B81F ,
```

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

```
505      "2B820 -> "2CEAF ,
```

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

```
506      "2CEB0 -> "2EBEF ,
```

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

```
507      "2EBF0 -> "2EE5F ,
```

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

```
508      "2F800 -> "2FA1F ,
```

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

```
509      "30000 -> "3134F ,
```

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

```
510      "31350 -> "323AF ,
```

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

```
511      "323B0 -> "3347F
```

```
512      }
```

`\c__xeCJK_CM_chars_clist` 包括日文假名浊点和异体字选择符。组合标识是最好是归入 256 类, 即透明类, 不会影响状态。但也会产生一定问题。比如下面的例子, 位于行尾的“二”造成分组不匹配。

```
\XeTeXinterchartokenstate=1
\XeTeXcharclass`二=256
\XeTeXinterchartoks 255 1 = {\bgroup}
\XeTeXinterchartoks 1 255 = {\egroup}
\XeTeXinterchartoks 1 1 = {x}
\font\zhfont="SimSun"
\zhfont
一二三二
\bye

513 \clist_const:Nn \c__xeCJK_CM_chars_clist
514 {
```

- Diacritics (音调符号)

```
515      "302A -> "302F ,
```

- 日文假名浊点

```
516      "3099 -> "309A ,
```

- Variation Selectors (异体字选择符)

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

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

```
518     "E0100 -> "E01EF
519 }
```

朝鲜文字母按 Unicode Hangul_Syllable_Type 分成初声(L)、中声(V)和终声(T)三类,以便区分音节内 shaping 与相邻音节边界(#158)。

```
520 \clist_const:Nn \c__xeCJK_HangulJamoL_chars_clist
521 { "1100 -> "115F , "A960 -> "A97C }
522 \clist_const:Nn \c__xeCJK_HangulJamoV_chars_clist
523 { "1160 -> "11A7 , "D7B0 -> "D7C6 }
524 \clist_const:Nn \c__xeCJK_HangulJamoT_chars_clist
525 { "11A8 -> "11FF , "D7CB -> "D7FB }
```

5.5 字符类别处理

\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。

```
526 \cs_new:Npn \xeCJK_class_num:n #1
527 { \use:c { \__xeCJK_class_csname:n {#1} } }
```

```
\xeCJKDeclareCharClass 528 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
529 {
530   \xeCJK_declare_char_class:nn {#2} {#3}
531   \IfBooleanT {#1} { \xeCJKResetPunctClass }
532 }
```

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。

```
533 \cs_new_protected:Npn \xeCJK_declare_char_class:nn #1#2
534 {
535   \clist_set:Nc \l__xeCJK_tmp_clist {#2}
536   \xeCJK_declare_char_class:nN {#1} \l__xeCJK_tmp_clist
537 }
538 \cs_new_protected:Npn \xeCJK_declare_char_class:nN #1#2
539 {
540   \clist_gconcat:ccN
541   { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } #2
542   \clist_map_inline:Nn #2
543   {
544     \str_if_eq:nnF {##1} { -> }
545     {
546       \__xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
547       { \xeCJK_class_num:n {#1} }
548     }
549   }
550   \xeCJK_set_char_class:nnn { "3099 } { "309A } { \xeCJK_class_num:n { CM } }
551 }
552 \NewDocumentCommand \__xeCJK_set_char_class_aux:Nnw
553 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
554 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
```

__xeCJK_check_num_range:nnNN 555 \cs_new_protected:Npn __xeCJK_check_num_range:nnNN #1#2#3#4

```
556 {
557   \bool_lazy_or:nnTF
558   { \tl_if_blank_p:n {#1} }
559   { \tl_if_blank_p:n {#2} }
560   {
561     \int_set:Nn #3 { \tl_if_blank:nTF {#1} {#2} {#1} }
562     \int_set_eq:NN #4 #3
563   }
564   {
565     \int_set:Nn #3 { \int_min:nn {#1} { \tl_if_novalue:nTF {#2} {#1} {#2} } }
566     \int_set:Nn #4 { \int_max:nn {#1} { \tl_if_novalue:nTF {#2} {#1} {#2} } }
567   }
568 }
```

```

569 \token_if_letter:NF ~~~~ac00
570 {
571   \int_set:Nn \l__xeCJK_begin_int { "AC00 }
572   \int_set:Nn \l__xeCJK_end_int { "D7A3 }
573   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
574   {
575     \char_set_catcode_letter:n { \l__xeCJK_begin_int }
576     \int_incr:N \l__xeCJK_begin_int
577   }
578 }

```

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

```

579 \cs_new_protected:Npn \xeCJK_set_char_class:nnn #1#2#3
580 {
581   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
582   \int_set:Nn \l__xeCJK_tmp_int {#3}
583   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
584   {
585     \tex_XeTeXcharclass:D \l__xeCJK_begin_int = \l__xeCJK_tmp_int
586     \int_incr:N \l__xeCJK_begin_int
587   }
588 }

```

`\__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。

```

589 \cs_new_protected:Npn \__xeCJK_set_char_class_eq:nn #1#2
590 {
591   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#2} }
592   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
593   { \tex_XeTeXcharclass:D ##1 = \l__xeCJK_tmp_int }
594 }

```

`\normalspacedchars` 声明前后不加间距的字符。

```

595 \NewDocumentCommand \normalspacedchars { m }
596 {
597   \tl_map_inline:nn {#1}
598   { \tex_XeTeXcharclass:D `##1 = \xeCJK_class_num:n { NormalSpace } }
599 }

```

`\xeCJKResetPunctClass` 用于重置标点符号所属的字符类。注意,重新声明 FullLeft 和 FullRight 类会覆盖 PoZheHaoLigature 与 LatinPunct 对共用标点的字符类调整,因此在重置后需要按记录的状态恢复这两个选项的设置。

```

600 \NewDocumentCommand \xeCJKResetPunctClass { }
601 {
602   \clist_gclear:N \g__xeCJK_HalfLeft_range_clist
603   \clist_gclear:N \g__xeCJK_HalfRight_range_clist
604   \clist_gclear:N \g__xeCJK_FullLeft_range_clist
605   \clist_gclear:N \g__xeCJK_FullRight_range_clist
606   \xeCJK_declare_char_class:nN { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
607   \xeCJK_declare_char_class:nN { HalfRight } \c__xeCJK_HalfRight_chars_clist
608   \xeCJK_declare_char_class:nN { FullLeft } \c__xeCJK_FullLeft_chars_clist
609   \xeCJK_declare_char_class:nN { FullRight } \c__xeCJK_FullRight_chars_clist
610   \bool_if:NT \l__xeCJK_pozhehao_ligature_bool
611   { \keys_set:nn { xeCJK / options } { PoZheHaoLigature = true } }
612   \bool_if:NT \l__xeCJK_latin_punct_bool
613   { \keys_set:nn { xeCJK / options } { LatinPunct = true } }
614 }
615 \bool_new:N \l__xeCJK_pozhehao_ligature_bool
616 \bool_new:N \l__xeCJK_latin_punct_bool

```

`\xeCJKResetCharClass` 用于恢复 xeCJK 对字符类别的设置。

```

617 \NewDocumentCommand \xeCJKResetCharClass { }
618 {

```

```

619 \clist_gclear:N \g__xeCJK_CJK_range_clist
620 \clist_gclear:N \g__xeCJK_NormalSpace_range_clist
621 \clist_gclear:N \g__xeCJK_CM_range_clist
622 \clist_gclear:N \g__xeCJK_HangulJamo_range_clist
623 \clist_gclear:N \g__xeCJK_HangulJamoL_range_clist
624 \clist_gclear:N \g__xeCJK_HangulJamoV_range_clist
625 \clist_gclear:N \g__xeCJK_HangulJamoT_range_clist
626 \clist_gclear:N \g__xeCJK_CJStarter_range_clist
627 \xeCJK_declare_char_class:nN { CJK } \c__xeCJK_CJK_chars_clist
628 \xeCJK_declare_char_class:nN { NormalSpace } \c__xeCJK_NormalSpace_chars_clist
629 \xeCJK_declare_char_class:nN { CM } \c__xeCJK_CM_chars_clist
630 \xeCJK_declare_char_class:nN { HangulJamoL } \c__xeCJK_HangulJamoL_chars_clist
631 \xeCJK_declare_char_class:nN { HangulJamoV } \c__xeCJK_HangulJamoV_chars_clist
632 \xeCJK_declare_char_class:nN { HangulJamoT } \c__xeCJK_HangulJamoT_chars_clist
633 \bool_if:NT \l__xeCJK_CJ_strict_bool
634 { \xeCJK_declare_char_class:nN { CJStarter } \c__xeCJK_CJ_chars_clist }
635 \xeCJKResetPunctClass
636 }
637 \bool_new:N \l__xeCJK_CJ_strict_bool

```

设置字符类别。

```
638 \xeCJKResetCharClass
```

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

```

639 \cs_new_protected:Npn \xeCJK_inter_class_toks:nnn #1#2#3
640 {
641   \tex_XeTeXinterchartoks:D \xeCJK_class_num:n {#1} ~
642                               \xeCJK_class_num:n {#2} = {#3}
643 }
644 \cs_generate_variant:Nn \xeCJK_inter_class_toks:nnn { nne }

```

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

```

645 \cs_new:Npn \xeCJK_get_inter_class_toks:nn #1#2
646 {
647   \tex_the:D \tex_XeTeXinterchartoks:D \xeCJK_class_num:n {#1} ~
648                               \xeCJK_class_num:n {#2}
649 }

```

`\xeCJK_clear_inter_class_toks:nn` 清除相邻类别之间的内容。注意, 直接赋空值可能会导致 $\text{\XeTeX}$ 崩溃。例如

```

\XeTeXinterchartokenstate = 1
\XeTeXcharclass`A=10
\XeTeXinterchartoks 10 10 = {xx}
\begingroup
  \XeTeXinterchartoks 10 10 = {} AA
\endgroup
\bye

```

如果把上述例子中的分组 `\begingroup` 和 `\endgroup` 去掉, 则结果正常, 甚为怪异。此处 $\text{\XeTeX}$ 的 bug 已经在 0.999992 版中修复¹⁴。

```

650 \cs_new_protected:Npn \xeCJK_clear_inter_class_toks:nn #1#2
651 { \xeCJK_inter_class_toks:nnn {#1} {#2} { \prg_do_nothing: } }

```

`\xeCJK_pre_inter_class_toks:nnn` 在相邻类别之间已有的内容前增加内容。

```

652 \cs_new_protected:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
653 {
654   \xeCJK_inter_class_toks:nne {#1} {#2}
655   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
656 }
657 \cs_generate_variant:Nn \xeCJK_pre_inter_class_toks:nnn { nne }

```

¹⁴<http://tug.org/svn/texlive?view=revision&revision=53880>

在相邻类别之间已有的内容后追加内容。

```
658 \cs_new_protected:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
659 {
660   \xeCJK_inter_class_toks:nne {#1} {#2}
661   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
662 }
663 \cs_generate_variant:Nn \xeCJK_app_inter_class_toks:nnn { nne }
```

将 #3 和 #4 之间的内容复制到 #1 和 #2 之间。

```
664 \cs_new_protected:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
665 {
666   \tl_set:Nc \l__xeCJK_tmp_tl
667   { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
668   \tl_if_empty:NTF \l__xeCJK_tmp_tl
669   {
670     \tl_set:Nc \l__xeCJK_tmp_tl
671     { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
672     \tl_if_empty:NF \l__xeCJK_tmp_tl
673     { \xeCJK_clear_inter_class_toks:nn {#1} {#2} }
674   }
675   { \xeCJK_inter_class_toks:nne {#1} {#2} { \exp_not:o \l__xeCJK_tmp_tl } }
676 }
```

将 #1 和 #2 之间出现的 #3 用 #4 替换。

```
677 \cs_new_protected:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
678 {
679   \tl_set:Nc \l__xeCJK_tmp_tl
680   { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
681   \tl_if_empty:NF \l__xeCJK_tmp_tl
682   {
683     \tl_replace_all:Nnn \l__xeCJK_tmp_tl {#3} {#4}
684     \xeCJK_inter_class_toks:nne {#1} {#2}
685     { \exp_not:o \l__xeCJK_tmp_tl }
686   }
687 }
```

清除边界与 CJK 文字、全角左右标点之间的内容。

```
688 \cs_new_protected:Npn \xeCJK_clear_Boundary_and_CJK_toks:
689 { }
690 \cs_new_protected:Npn \__xeCJK_update_clear_toks:n #1
691 {
692   \cs_gset_protected:Npe \xeCJK_clear_Boundary_and_CJK_toks:
693   {
694     \exp_not:o { \xeCJK_clear_Boundary_and_CJK_toks: }
695     \tex_XeTeXinterchartoks:D
696     \xeCJK_class_num:n { Boundary } ~
697     \xeCJK_class_num:n {#1} = { \exp_not:N \prg_do_nothing: }
698   }
699 }
```

保存宏包预先定义的字符类。

```
700 \seq_new:N \g__xeCJK_base_class_seq
701 \seq_gset_eq:NN \g__xeCJK_base_class_seq \g__xeCJK_class_seq
702 \seq_new:N \g__xeCJK_non_CJK_class_seq
703 \seq_gset_from_clist:Nn \g__xeCJK_non_CJK_class_seq
704 { Default , HalfLeft , HalfRight , NormalSpace , Boundary }
705 \seq_new:N \g__xeCJK_CJK_class_seq
706 \cs_new_protected:Npn \__xeCJK_save_CJK_class:n #1
707 {
708   \seq_gput_right:Nn \g__xeCJK_CJK_class_seq {#1}
709   \tl_const:cn
710   { \__xeCJK_CJK_class_tl:n { \use:c { \__xeCJK_class_csname:n {#1} } } }
711   {#1}
712   \__xeCJK_update_clear_toks:n {#1}
```

```

713 }
714 \clist_map_function:nN
715 {
716   CJK , FullLeft , FullRight , CM , HangulJamo ,
717   HangulJamoL , HangulJamoV , HangulJamoT , CJStarter , PoZheHao
718 }
719 \__xeCJK_save_CJK_class:n

```

5.6 字符输出规则

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

\xeCJK_class_group_begin: 在 CJK 类开始时, 设置 \XeTeXdashbreakstate 为零, 避免破折号之间的折行。
 \xeCJK_class_group_end:

```

720 \cs_new_protected:Npn \xeCJK_class_group_begin:
721 {
722   \c_group_begin_token
723   \bool_set_true:N \l__xeCJK_CJK_group_bool
724   \xeCJK_reset_space_factor:
725   \int_zero:N \tex_XeTeXdashbreakstate:D
726 }
727 \bool_new:N \l__xeCJK_CJK_group_bool
728 \cs_new_eq:NN \xeCJK_class_group_end: \c_group_end_token

```

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

```

729 \AtEndOfPackage
730 {
731   \seq_map_inline:Nn \g__xeCJK_class_seq
732   {
733     \str_if_eq:nnTF {#1} { CM }
734     { \xeCJK_copy_inter_class_toks:nnnn { CM } {#1} { CJK } { CJK } }
735     {
736       \xeCJK_copy_inter_class_toks:nnnn { CM } {#1} { CJK } {#1}
737       \str_if_eq:nnF {#1} { CJK }
738       { \xeCJK_copy_inter_class_toks:nnnn {#1} { CM } {#1} { CJK } }
739     }
740   }
741 }

```

HangulJamo 字符类与 CJK 字符类基本相同, 只是 HangulJamo 类之间不加入任何内容。

```

742 \AtEndOfPackage
743 {
744   \seq_map_inline:Nn \g__xeCJK_class_seq
745   {
746     \str_if_eq:nnF {#1} { HangulJamo }
747     {
748       \xeCJK_copy_inter_class_toks:nnnn { HangulJamo } {#1} { CJK } {#1}
749       \xeCJK_copy_inter_class_toks:nnnn {#1} { HangulJamo } {#1} { CJK }
750     }
751   }
752 }

```

HangulJamoL、HangulJamoV、HangulJamoT 对外都复制 CJK 的转移规则。三类之间先复制 CJK 到 CJK, 再按 UAX #29 的 Hangul 音节延续规则清空 L-L、L-V、V-V、V-T、T-T 五种转移; 其余组合是音节边界, 保留 \CJKglue 和断行机会。

```

753 \AtEndOfPackage
754 {
755   \clist_map_inline:nn { HangulJamoL , HangulJamoV , HangulJamoT }
756   {
757     \seq_map_inline:Nn \g__xeCJK_class_seq
758     {
759       \str_if_eq:nnF {##1} {#1}
760       {
761         \xeCJK_copy_inter_class_toks:nnnn {#1} {##1} { CJK } {##1}
762         \xeCJK_copy_inter_class_toks:nnnn {##1} {#1} {##1} { CJK }
763       }
764     }
765   }
766   \clist_map_inline:nn { HangulJamoL , HangulJamoV , HangulJamoT }
767   {
768     \clist_map_inline:nn { HangulJamoL , HangulJamoV , HangulJamoT }
769     { \xeCJK_copy_inter_class_toks:nnnn {#1} {##1} { CJK } { CJK } }
770   }
771   \xeCJK_clear_inter_class_toks:nn { HangulJamoL } { HangulJamoL }
772   \xeCJK_clear_inter_class_toks:nn { HangulJamoL } { HangulJamoV }
773   \xeCJK_clear_inter_class_toks:nn { HangulJamoV } { HangulJamoV }
774   \xeCJK_clear_inter_class_toks:nn { HangulJamoV } { HangulJamoT }
775   \xeCJK_clear_inter_class_toks:nn { HangulJamoT } { HangulJamoT }
776 }

```

CJStarter 对外复制 CJK 转移, 并在普通 CJK、同类小假名以及全角右标点之后进入该类时, 把禁则 penalty 放在 \CJKglue 前。字符只有在 CJLineBreak=strict 时才归入该类。

```

777 \AtEndOfPackage
778 {
779   \seq_map_inline:Nn \g__xeCJK_class_seq
780   {
781     \str_if_eq:nnF {#1} { CJStarter }
782     {
783       \xeCJK_copy_inter_class_toks:nnnn { CJStarter } {#1} { CJK } {#1}
784       \xeCJK_copy_inter_class_toks:nnnn {#1} { CJStarter } {#1} { CJK }
785     }
786   }
787   \xeCJK_copy_inter_class_toks:nnnn
788   { CJStarter } { CJStarter } { CJK } { CJK }
789   \xeCJK_pre_inter_class_toks:nnn { CJK } { CJStarter } { \xeCJK_no_break: }
790   \xeCJK_pre_inter_class_toks:nnn
791   { CJStarter } { CJStarter } { \xeCJK_no_break: }
792   \xeCJK_inter_class_toks:nnn { FullRight } { CJStarter }
793   {
794     \xeCJK_FullRight_and_CJStarter:
795     \xeCJK_fallback_symbol:NN
796     \CJKsymbol
797   }
798 }
799 \cs_new_protected:Npn \xeCJK_FullRight_and_CJStarter:
800 {
801   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
802   \xeCJK_no_break:
803   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
804   \__xeCJK_select_font:
805   \CJKglue
806 }

```

PoZheHao 字符类与 FullRight 字符类基本相同, 只是 PoZheHao 类之间不加入任何内容, 从而不阻断 OpenType 字体的破折号合字功能(#382)。

```

807 \AtEndOfPackage

```

```

808 {
809   \seq_map_inline:Nn \g__xeCJK_class_seq
810   {
811     \str_if_eq:nnF {#1} { PoZheHao }
812     {
813       \xeCJK_copy_inter_class_toks:nnnn { PoZheHao } {#1} { FullRight } {#1}
814       \xeCJK_copy_inter_class_toks:nnnn {#1} { PoZheHao } {#1} { FullRight }
815     }
816   }
817 }

818 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
819 {
820   \xeCJK_inter_class_toks:nnn {#1} { CJK }
821   {
822     \__xeCJK_boundary_capture_class:n { CJK }
823     \xeCJK_class_group_begin:
824     \xeCJK_select_font:
825     \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
826     \xeCJK_clear_Boundary_and_CJK_toks:
827     \xeCJK_fallback_symbol:NN
828     \CJKsymbol
829   }
830   \xeCJK_inter_class_toks:nnn { CJK } {#1}
831   {
832     \xeCJK_class_group_end:
833     \__xeCJK_boundary_capture_class:n { default }
834   }
835 }

836 \clist_map_inline:nn { Default , HalfLeft }
837 {
838   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
839   { \xeCJK_Boundary_and_Default: }
840   \xeCJK_app_inter_class_toks:nnn { CJK } {#1}
841   { \CJKecglue }
842 }

```

字体组、颜色组或其他命令结束后，源码空格可能已经生成普通词间 glue，压在仍然有效的 CJK marker 上方。只有已知命令或分组路径设置了 `\g__xeCJK_glue_check_pending_bool` 时，才允许暂时移除末尾 glue 并检查其下方的 marker，避免把直接输入的显式 `\hskip` 当成源码空格。这枚 glue 还必须 finite、带 shrink 且自然宽度与当前词间空格相等；对未进入额外句末空白分支的普通词间 glue，TeX 会把 stretch 乘以 `\spacefactor/1000`、把 shrink 乘以 `1000/\spacefactor`，所以不能要求整份 skip 严格相等。当 `\spacefactor >= 2000` 时还会把 `\fontdimen7` 加到自然宽度，自然宽度检查会排除这种情况。只有下方是 CJK、CJK-space 或 CJK-widow marker 时，才移除这枚 glue，并换成缓存的 `\CJKecglue`；否则原样放回。移除前用 skip 变量完整保存 `\lastskip`，restore 还回的是同一份 natural/stretch/shrink 数值，不会按当前 spacefactor 重新计算；只有节点的源码来源无法恢复。节点列表无法区分源码空格与参数完全相同的显式 `\hskip`。因此，紧跟在已注册命令之后、自然宽度等于当前词间空格且带 shrink 的显式 glue 也可能被当作源码空格。需要保留它时，可在前面加零宽 `\kern`，或者改用不同的自然宽度、去掉 shrink，让检查不会继续向前找到 xeCJK 的 marker。restore/fallback 不消除 `\g__xeCJK_glue_check_pending_bool`：该 glue 可能来自 ulem 内部列表，这个布尔状态还要留给命令外真正的 CJK 边界；只有确认并使用 marker 后才结束 pending 状态。fallback 也不继续调用 `\__xeCJK_check_for_ecglue_aux:`。此时候选 glue 仍是 last node（或已由 restore 原样还回），aux 无法越过它取得下方 marker；这枚没有通过检查的 glue 本身就是要保留的分隔。只有 success 路径确认下方是 CJK marker 并移除 glue 后，才允许改发 `\CJKecglue`。

`\g__xeCJK_glue_check_pending_bool` 是全局布尔变量。在盒子末尾结束 catcode-2 分组时设为 true 后，它可能越过 `\setbox`，一直留到下一个水平列表。pending=true 只表示“下一次可以检查紧跟在 xeCJK marker 后的源码空格”。空列表中不可能有这样的 marker，所以顶层恢复链（没有活跃 capture）发现 `\tex_lastnodetype:D` 为负时，会直接清除 pending 状

态。capture 活跃时不这样做:ulem 等 stream 会在内部固定宽度盒子中触发同一恢复链,此时仍需把 pending 留给命令外的边界;capture 入口本身已经清空旧状态。

```

843 \cs_new_protected:Npn \xeCJK_Boundary_and_Default:
844 {
845   \__xeCJK_boundary_capture_class:n { default }
846   \xeCJK_check_for_ecglue:
847 }
848 \cs_new_protected:Npn \__xeCJK_check_for_xecglue:
849 {
850   \__xeCJK_if_last_glue:TF
851   { \__xeCJK_replace_space: }
852   { \__xeCJK_check_for_ecglue: }
853 }
854 \cs_new_protected:Npn \__xeCJK_check_for_ecglue:
855 {
856   \__xeCJK_if_last_glue:TF
857   { \__xeCJK_recover_ecglue_source_space: }
858   { \__xeCJK_check_for_ecglue_aux: }
859 }
860 \cs_new_protected:Npn \__xeCJK_check_for_ecglue_aux:
861 {
862   \xeCJK_if_last_node:nTF { CJK }
863   { \use_i:nn }
864   { \xeCJK_if_last_node:nTF { CJK-widow } }
865   { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
866   {
867     \xeCJK_if_last_node:nTF { CJK-space }
868     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
869     {
870       \xeCJK_if_last_node:nTF { math }
871       { \xeCJK_remove_node: }
872       {
873         \__xeCJK_boundary_if_last_math_space:TF
874         { \__xeCJK_boundary_math_space_consume: }
875         { \__xeCJK_glue_check_expire_stale: }
876       }
877     }
878   }
879 }
880 \dim_new:N \l__xeCJK_interword_dim
881 \prg_new_conditional:Npnn \__xeCJK_skip_if_interword:N #1 { T , F , TF }
882 {
883   \skip_if_finite:nTF {#1}
884   {
885     \dim_set:Nn \l__xeCJK_tmp_dim {#1}
886     \skip_set:Nn \l__xeCJK_tmp_skip { \c_xeCJK_space_skip_tl }
887     \dim_set:Nn \l__xeCJK_interword_dim { \l__xeCJK_tmp_skip }
888     \dim_compare:nNnTF
889     { \l__xeCJK_tmp_dim } = { \l__xeCJK_interword_dim }
890     {
891       \dim_compare:nNnTF
892       { \tex_glueshrink:D #1 } > \c_zero_dim
893       { \prg_return_true: }
894       { \prg_return_false: }
895     }
896     { \prg_return_false: }
897   }
898   { \prg_return_false: }
899 }
900 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space:
901 {
902   \bool_if:NTF \g__xeCJK_glue_check_pending_bool
903   {
904     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
905     \__xeCJK_skip_if_interword:N \l__xeCJK_last_skip
906     {

```

```

907         \tex_unskip:D
908         \__xeCJK_recover_ecglue_source_space_aux:
909     }
910     { \__xeCJK_recover_ecglue_source_space_fallback: }
911 }
912 { \__xeCJK_recover_ecglue_source_space_fallback: }
913 }
914 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_aux:
915 {
916     \xeCJK_if_last_node:nTF { CJK }
917     { \__xeCJK_recover_ecglue_source_space_success: }
918     {
919         \xeCJK_if_last_node:nTF { CJK-widow }
920         { \__xeCJK_recover_ecglue_source_space_success: }
921         {
922             \xeCJK_if_last_node:nTF { CJK-space }
923             { \__xeCJK_recover_ecglue_source_space_success: }
924             {
925                 \xeCJK_if_last_node:nTF { math }
926                 {
927                     \xeCJK_remove_node:
928                     \__xeCJK_recover_ecglue_source_space_restore:
929                     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
930                 }
931                 {
932                     \__xeCJK_boundary_if_last_math_space:TF
933                     { \__xeCJK_boundary_math_space_consume: }
934                     { \__xeCJK_recover_ecglue_source_space_restore: }
935                 }
936             }
937         }
938     }
939 }
940 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_success:
941 {
942     \xeCJK_remove_node:
943     \skip_horizontal:N \l__xeCJK_ecglue_skip
944     \bool_if:NT \g__xeCJK_glue_check_pending_bool
945     { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
946 }
947 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_restore:
948 {
949     \skip_horizontal:N \l__xeCJK_last_skip
950 }
951 \cs_new_eq:NN \__xeCJK_recover_ecglue_source_space_fallback: \prg_do_nothing:
952 \cs_new_protected:Npn \__xeCJK_glue_check_expire_stale:
953 {
954     \int_compare:nNnT { \g__xeCJK_boundary_capture_depth_int } = \c_zero_int
955     {
956         \int_compare:nNnT { \tex_lastnodetype:D } < \c_zero_int
957         { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
958     }
959 }
960 \cs_new_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:

```

`\__xeCJK_replace_space:` 将空格替换为 `\CJKecglue`。注意由 `\leaders` 等产生的 `glue`，并不能正确地还回去。好在 $\text{\LaTeX} 2_{\epsilon}$ 中常用的 `\hrulefill` 和 `\dotfill` 定义末尾都有 `\kern\z@` 保护。

```

961 \cs_new_protected:Npn \__xeCJK_replace_space:
962 {
963     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
964     \tex_unskip:D
965     \xeCJK_if_last_node:nTF { CJK-space }
966     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
967     {
968         \xeCJK_if_last_node:nTF { CJK }
969         {

```

```

970         \skip_if_eq:nnTF
971         { \l__xeCJK_last_skip }
972         { \c_xeCJK_space_skip_tl }
973         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
974         { \skip_horizontal:N \l__xeCJK_last_skip }
975     }
976     {
977         \__xeCJK_boundary_if_last_math_space:TF
978         {
979             \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
980             { \__xeCJK_boundary_math_space_consume: }
981             { \skip_horizontal:N \l__xeCJK_last_skip }
982         }
983         { \skip_horizontal:N \l__xeCJK_last_skip }
984     }
985 }
986 }
987 \skip_new:N \l__xeCJK_last_skip

988 \clist_map_inline:nn { Default , HalfRight }
989 {
990     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
991     {
992         \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
993         \peek_meaning_remove:NTF \tex_italiccorrection:D
994         {
995             \tex_italiccorrection:D
996             { \xeCJK_make_node:n { default } }
997         }
998         {
999             \token_if_space:NTF \l_peek_token
1000             { \xeCJK_make_space_node: }
1001             { { \xeCJK_make_node:n { default } } }
1002         }
1003     }
1004     \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue }
1005 }

1006 \xeCJK_inter_class_toks:nnn { Boundary } { NormalSpace }
1007 { \xeCJK_Boundary_and_NormalSp: }

\xeCJK_Boundary_and_NormalSp: 1008 \cs_new_protected:Npn \xeCJK_Boundary_and_NormalSp:
1009 { \xeCJK_check_for_ecglue_normalsp: }
1010 \cs_new_protected:Npn \__xeCJK_check_for_xecglue_normalsp:
1011 {
1012     \__xeCJK_if_last_glue:TF
1013     { \__xeCJK_replace_space: }
1014     { \__xeCJK_check_for_ecglue_normalsp: }
1015 }
1016 \cs_new_protected:Npn \__xeCJK_check_for_ecglue_normalsp:
1017 {
1018     \xeCJK_if_last_node:nT { CJK-space }
1019     { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
1020 }
1021 \cs_new_eq:NN \xeCJK_check_for_ecglue_normalsp:
1022     \__xeCJK_check_for_ecglue_normalsp:

1023 \xeCJK_inter_class_toks:nnn { NormalSpace } { Boundary }
1024 {
1025     \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
1026     \peek_meaning_remove:NTF \tex_italiccorrection:D
1027     {
1028         \tex_italiccorrection:D
1029         { \xeCJK_make_node:n { normalspace } }
1030     }
1031     {
1032         \token_if_space:NTF \l_peek_token
1033         { \xeCJK_make_space_node: }

```

```

1034         { { \xeCJK_make_node:n { normalspace } } }
1035     }
1036 }

1037 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
1038 {
1039     \__xeCJK_boundary_capture_class:n { CJK }
1040     \xeCJK_check_for_glue:
1041     \xeCJK_class_group_begin:
1042     \xeCJK_clear_Boundary_and_CJK_toks:
1043     \xeCJK_select_font:
1044     \xeCJK_fallback_symbol:NN
1045     \CJKsymbol
1046 }

\xeCJK_check_for_glue: 1047 \cs_new_protected:Npn \xeCJK_check_for_glue:
1048 {
1049     \__xeCJK_if_last_kern:TF
1050     { \__xeCJK_check_for_glue_auxi: }
1051     {
1052         \__xeCJK_if_last_glue:TF
1053         { \__xeCJK_check_for_glue_skip: }
1054         {
1055             \__xeCJK_if_last_math:TF
1056             { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1057             {
1058                 \__xeCJK_glue_check_expire_stale:
1059                 \__xeCJK_check_for_glue_auxii:
1060             }
1061         }
1062     }
1063 }

```

统一 framework 在已知命令出口重放 xeCJK marker, 并通过 `\g_@@_glue_check_pending_bool` 开启一次源码空格检查; 显式分组的 `catcode 2` 结束点也设置同一个布尔状态。若待检查的 glue 下方不是 kern marker, 本函数原样还回 glue 并进入标点检测链, 不再根据任意 hlist、whatsit 或可能陈旧的 `\g_@@_last_node_tl` 猜测输出类别。

这枚 glue 使用与 Boundary→Default 方向相同的 `\__xeCJK_skip_if_interword:N` 判断函数检查: listings 等宏包在 CJK 字符间放置的 fil 级列对齐 glue 不是 finite; 标题格式中 `\quad` 等有意放置的间距没有 shrink 分量; 显式 `\hskip` 的自然宽度通常不等于当前词间空格。这三种情况都不进入源码空格分支。如果显式 glue 在节点列表中与词间空格没有任何区别, TeX 仍无法判断它的来源; 需要原样保留时, 与 Default 方向一样, 在前面加零宽 `\kern`。

本函数仅在 `\@@_if_last_glue:TF` 的 true 分支中调用, 因此入口处 `\tex_lastskip:D` 保证为非零 glue。

```

1064 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip:
1065 {
1066     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1067     \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
1068     {
1069         \tex_unskip:D
1070         \__xeCJK_if_last_kern:TF
1071         { \__xeCJK_check_for_glue_skip_kern: }
1072         { \__xeCJK_check_for_glue_skip_nonkern: }
1073     }
1074     { \__xeCJK_check_for_glue_skip_consume_and_fallback: }
1075 }
1076 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_kern:
1077 {
1078     \__xeCJK_boundary_if_last_math_space:TF
1079     {
1080         \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1081         { \__xeCJK_boundary_math_space_to_CJK: }
1082         { \__xeCJK_check_for_glue_skip_restore: }
1083     }

```

```

1084     { \_xeCJK\_check\_for\_glue\_skip\_kern\_normal: }
1085   }
1086 \cs\_new\_protected:Npn \_xeCJK\_check\_for\_glue\_skip\_kern\_normal:
1087 {
1088   \dim\_compare:nNnTF \tex\_lastkern:D = { \_xeCJK\_node:n { math } }
1089   {
1090     \bool\_if:NNTF \g\_xeCJK\_glue\_check\_pending\_bool
1091     {
1092       \xeCJK\_remove\_node:
1093       \bool\_gset\_false:N \g\_xeCJK\_glue\_check\_pending\_bool
1094       \bool\_if:NNTF \l\_xeCJK\_xecglue\_bool
1095       { \skip\_horizontal:N \l\_xeCJK\_ecglue\_skip }
1096       { \skip\_horizontal:N \l\_xeCJK\_last\_skip }
1097     }
1098     { \_xeCJK\_check\_for\_glue\_skip\_restore: }
1099   }
1100   {
1101     \bool\_if:NNTF \g\_xeCJK\_glue\_check\_pending\_bool
1102     {
1103       \bool\_gset\_false:N \g\_xeCJK\_glue\_check\_pending\_bool
1104       \dim\_case:nnF { \tex\_lastkern:D }
1105       {
1106         { \_xeCJK\_node:n { CJK } }
1107         {
1108           \xeCJK\_remove\_node:
1109           \bool\_if:NNTF \l\_xeCJK\_reserve\_space\_bool
1110           { \skip\_horizontal:N \l\_xeCJK\_last\_skip }
1111           { \CJKglue }
1112         }
1113         { \_xeCJK\_node:n { CJK-space } }
1114         { \xeCJK\_remove\_node: \_xeCJK\_ccglue\_or\_space: }
1115         { \_xeCJK\_node:n { CJK-widow } }
1116         {
1117           \xeCJK\_remove\_node:
1118           \xeCJK\_widow\_penalty:
1119           \bool\_if:NNTF \l\_xeCJK\_reserve\_space\_bool
1120           { \skip\_horizontal:N \l\_xeCJK\_last\_skip }
1121           { \CJKglue }
1122         }
1123       }
1124       { \_xeCJK\_check\_for\_glue\_skip\_restore: }
1125     }
1126     { \_xeCJK\_check\_for\_glue\_skip\_restore: }
1127   }
1128 }
1129 \cs\_new\_protected:Npn \_xeCJK\_check\_for\_glue\_skip\_nonkern:
1130 {
1131   \bool\_if:NT \g\_xeCJK\_glue\_check\_pending\_bool
1132   { \bool\_gset\_false:N \g\_xeCJK\_glue\_check\_pending\_bool }
1133   \_xeCJK\_if\_last\_math:TF
1134   {
1135     \bool\_if:NNTF \l\_xeCJK\_xecglue\_bool
1136     { \skip\_horizontal:N \l\_xeCJK\_ecglue\_skip }
1137     { \skip\_horizontal:N \l\_xeCJK\_last\_skip }
1138   }
1139   { \_xeCJK\_check\_for\_glue\_skip\_restore: }
1140 }
1141 \cs\_new\_protected:Npn \_xeCJK\_check\_for\_glue\_skip\_restore:
1142 {
1143   \skip\_horizontal:N \l\_xeCJK\_last\_skip
1144   \_xeCJK\_check\_for\_glue\_auxii:
1145 }
1146 \cs\_new\_protected:Npn \_xeCJK\_check\_for\_glue\_skip\_consume\_and\_fallback:
1147 {
1148   \bool\_if:NT \g\_xeCJK\_glue\_check\_pending\_bool
1149   { \bool\_gset\_false:N \g\_xeCJK\_glue\_check\_pending\_bool }
1150   \_xeCJK\_check\_for\_glue\_auxii:

```

```

1151 }
1152 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxi:
1153 {
1154   \dim_case:nn { \tex_lastkern:D }
1155   {
1156     { \__xeCJK_node:n { CJK } }
1157     { \xeCJK_remove_node: \CJKglue }
1158     { \__xeCJK_node:n { CJK-space } }
1159     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1160     { \__xeCJK_node:n { CJK-widow } }
1161     { \xeCJK_remove_node: \xeCJK_widow_penalty: \CJKglue }
1162     { \__xeCJK_node:n { default } }
1163     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1164     { \__xeCJK_node:n { math } }
1165     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1166     { \__xeCJK_node:n { math-space } }
1167     { \__xeCJK_boundary_math_space_to_CJK: }
1168     { \__xeCJK_node:n { math-space-frozen } }
1169     { \__xeCJK_boundary_math_space_to_CJK: }
1170   }
1171 }
1172 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxii:
1173 {
1174   \xeCJK_if_last_punct:TF
1175   { \__xeCJK_check_for_glue_auxiii: }
1176   { \xeCJK_check_for_xglue: }
1177 }
1178 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxiii:
1179 {
1180   \bool_if:NT \l__xeCJK_last_penalty_bool
1181   { \tex_penalty:D \l__xeCJK_last_penalty_int }
1182   \skip_horizontal:N \l__xeCJK_last_skip
1183   \tl_if_eq:NNF \l__xeCJK_aligni_tl \c__xeCJK_left_tl { \CJKglue }
1184 }
1185 \cs_new_eq:NN \xeCJK_check_for_xglue: \prg_do_nothing:
1186 \cs_new_protected:Npn \__xeCJK_check_for_xglue:
1187 {
1188   \__xeCJK_if_last_glue:T
1189   {
1190     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1191     \tex_unskip:D
1192     \xeCJK_if_last_node:nTF { CJK-space }
1193     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1194     {
1195       \xeCJK_if_last_node:nTF { default-space }
1196       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1197       { \__xeCJK_check_for_xglue_aux: }
1198     }
1199   }
1200 }
1201 \cs_new_protected:Npn \__xeCJK_check_for_xglue_aux:
1202 {
1203   \skip_if_eq:nnTF
1204   { \l__xeCJK_last_skip }
1205   { \c__xeCJK_space_skip_tl }
1206   {
1207     \xeCJK_if_last_node:nTF { CJK }
1208     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1209     {
1210       \xeCJK_if_last_node:nTF { default }
1211       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1212       {
1213         \xeCJK_if_last_node:nTF { math }
1214         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1215         {
1216           \__xeCJK_if_last_math:TF
1217           { \skip_horizontal:N \l__xeCJK_ecglue_skip }

```

```

1218             { \skip_horizontal:N \l__xeCJK_last_skip }
1219         }
1220     }
1221 }
1222 }
1223 { \skip_horizontal:N \l__xeCJK_last_skip }
1224 }
1225 \cs_new_protected:Npn \__xeCJK_ccglue_or_space:
1226 { \CJKglue }

```

__xeCJK_if_last_none:TF 一些 ϵ -TeX 结点判定函数。

```

\underline{\__xeCJK_if_last_hlist:TF}
\underline{\__xeCJK_if_last_math:TF}
\underline{\__xeCJK_if_last_glue:TF}
\underline{\__xeCJK_if_last_kern:TF}
\underline{\__xeCJK_if_last_penalty:TF}
1227 \group_begin:
1228 \cs_set:Npn \__xeCJK_tmp:nn #1
1229 {
1230     \exp_args:Ncc \__xeCJK_tmp_aux:NNn
1231     { \__xeCJK_if_last_ #1 : }
1232     { c__xeCJK_ #1 _node }
1233 }
1234 \cs_set:Npn \__xeCJK_tmp_aux:NNn #1#2#3
1235 {
1236     \int_const:Nn #2 {#3}
1237     \prg_new_conditional:Npnn #1 { T , F , TF }
1238     {
1239         \if_int_compare:w \tex_lastnodetype:D = #2
1240             \prg_return_true: \else: \prg_return_false: \fi:
1241     }
1242 }
1243 \__xeCJK_tmp:nn { none } { -1 }
1244 \__xeCJK_tmp:nn { hlist } { 1 }
1245 \__xeCJK_tmp:nn { math } { 10 }
1246 \__xeCJK_tmp:nn { glue } { 11 }
1247 \__xeCJK_tmp:nn { kern } { 12 }
1248 \__xeCJK_tmp:nn { penalty } { 13 }
1249 \__xeCJK_tmp:nn { whatsit } { 9 }
1250 \group_end:

\underline{\xeCJK_if_last_node_p:n}
\underline{\xeCJK_if_last_node:nTF}
1251 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
1252 {
1253     \if_dim:w
1254         \cs_if_exist_use:cTF { c__xeCJK_#1_node_dim }
1255         { = \tex_lastkern:D }
1256         { \use:c { c__xeCJK_#1_node_skip } = \tex_lastskip:D }
1257         \prg_return_true: \else: \prg_return_false: \fi:
1258     }

```

\xeCJK_declare_node:n 用于判断插入的各种 kern 和 glue。

```

\underline{\xeCJK_make_node:n}
1259 \cs_new_protected:Npn \xeCJK_declare_node:n #1
1260 {
1261     \int_gincr:N \g__xeCJK_node_int
1262     \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
1263     { \dim_gset:cn } { \dim_const:cn }
1264     { c__xeCJK_#1_node_dim } { \g__xeCJK_node_int sp }
1265 }
1266 \cs_new_protected:Npn \xeCJK_declare_glue_node:n #1
1267 {
1268     \int_gincr:N \g__xeCJK_node_int
1269     \skip_if_exist:cTF { c__xeCJK_#1_node_skip }
1270     { \skip_gset:cn } { \skip_const:cn }
1271     { c__xeCJK_#1_node_skip } { \g__xeCJK_node_int sp }
1272 }
1273 \int_new:N \g__xeCJK_node_int
1274 \int_gset:Nn \g__xeCJK_node_int { 10 }
1275 \tl_new:N \g__xeCJK_last_node_tl
1276 \cs_new_protected:Npn \xeCJK_make_node:n #1
1277 {

```

```

1278 \exp_args:Nc \__xeCJK_make_node:N { c__xeCJK_#1_node_dim }
1279 \tl_gset:Nn \g__xeCJK_last_node_tl {#1}
1280 }
1281 \cs_new:Npn \__xeCJK_node:n #1
1282 { \use:c { c__xeCJK_#1_node_dim } }
1283 \cs_new:Npn \__xeCJK_glue_node:n #1
1284 { \use:c { c__xeCJK_#1_node_skip } }
1285 \cs_new_protected:Npn \__xeCJK_make_node:N #1
1286 {
1287   \tex_kern:D - #1
1288   \tex_kern:D #1
1289 }
1290 \cs_new_protected:Npn \xeCJK_remove_node:
1291 {
1292   \__xeCJK_if_last_kern:TF
1293   { \tex_unkern:D \tex_unkern:D }
1294   {
1295     \__xeCJK_if_last_glue:T
1296     { \tex_unskip:D \tex_unskip:D }
1297   }
1298   \tl_gclear:N \g__xeCJK_last_node_tl
1299 }
1300 \xeCJK_declare_node:n { CJK }
1301 \xeCJK_declare_node:n { CJK-space }
1302 \xeCJK_declare_node:n { default }
1303 \xeCJK_declare_node:n { CJK-widow }
1304 \xeCJK_declare_node:n { normalspace }
1305 \xeCJK_declare_glue_node:n { default-space }
1306 \xeCJK_declare_node:n { math }
1307 \xeCJK_declare_node:n { math-space }
1308 \xeCJK_declare_node:n { math-space-frozen }

1309 \bool_new:N \g__xeCJK_glue_check_pending_bool

```

命令边界框架提供五类注册策略。box 策略在注册命令执行前保存并移除前一可见字符留下的 marker 和可选源码空格, 命令结束后取出其末尾 hbox, 重建边界后原样放回; wrapped-box 用透明 hbox 收集会直接写出多个节点的盒子命令。stream 策略让 annotation、数学包装或 verbatim 流直接写入当前列表; 其 default 变体固定两端为 Default, first-default 只固定首类别、仍以实际输出决定末类别。transparent 策略用于只插入锚点、write 等无可见输出的命令。其中 \null 还能合法地出现在 \setbox(box) 后, 不能在其前面插入 hook 代码, 故使用 post-transparent after-only 变体: 只在宽、高、深均为零的盒子确实盖住 marker 或“marker + 候选 glue”时, 以列表中的真实 marker 为证据, 把这一边界后缀移到盒子后; 其下方没有 marker 时按原顺序还原 glue 与盒子, 不能把任意 glue 搬过 hbox。Boundary 到 Default/CJK 的转换会把观察到的类别同时写入所有活跃 capture, 因而混合输出和嵌套命令都能分别取得实际首、尾类别。

capture 结束时, 框架按“前类别 × 首类别 × 是否有源码空格”重建左边界, 再按实际末类别重放 marker 供右边界使用; 无可见输出时则完整恢复入口状态。Default 字符在盒子内部设置的 \spacefactor 不会传播到外层列表, 故重放 Default 类 marker 时还要把源码空格检查使用的值同步为外层列表当前的 \spacefactor; stream 内容会直接更新外层值, 因此使用同一条同步路径。“无可见输出”由观察类别、盒子尺寸与末节点类型共同判定。 \vrule 不触发 interchar 转换; 若盒子宽度非零、高度或深度非零, 且末节点是这个盒子直接排出的 char、rule 或标点处理添加的 kern, box/wrapped-box 就按 Default 首尾重建边界。行内公式另用 math 类别: stream 和 box 的正文以 \$、\ (或 \ensuremath 开始时, 参数适配器必须在 math-on 节点之前报告首类别; 正文语法以相应公式结束时只产生候选, 正文实际排完后还要由末尾 math 节点或 marker 确认, 才把 math 记为当前 capture 的确定末类别。未知宏若把公式记号或分组当作参数消费, 实际节点检查不会通过。capture 结束时以这个明确的正文末类别覆盖命令内部留下的旧 marker, 不把任意内部 math 节点猜成可见公式。这样“CJK 前缀 + 公式后缀”和外层分组中的公式都能重放正确的右边界, 同时嵌套命令只把结果逐层写回其实际输出列

```

\__xeCJK_boundary_capture_class:n
\__xeCJK_boundary_if_math_head:n
\__xeCJK_boundary_if_math_tail:n
\__xeCJK_boundary_if_math_tail_space:n
\__xeCJK_boundary_if_math_edge:n
\__xeCJK_boundary_if_last_math_node:
\__xeCJK_boundary_math_begin:
\__xeCJK_boundary_math_begin_visible:
\__xeCJK_boundary_math_end:n
\__xeCJK_boundary_math_end_confirm_node:
\__xeCJK_boundary_math_end_confirm_space:
\__xeCJK_boundary_math_end_confirm_space_visible:
\__xeCJK_boundary_math_end_confirm:
\__xeCJK_boundary_capture_report_first:n
\__xeCJK_boundary_capture_allocate:n
\__xeCJK_boundary_capture_suspend:
\__xeCJK_boundary_capture_resume:
\__xeCJK_boundary_capture_begin:
\__xeCJK_boundary_inline_box_begin:
\__xeCJK_boundary_inline_box_end:n
\__xeCJK_boundary_inline_last_box_begin:
\__xeCJK_boundary_inline_last_box_end:n
\__xeCJK_boundary_inline_stream_begin:n
\__xeCJK_boundary_inline_stream_begin:nn
\__xeCJK_boundary_inline_stream_end:n
\__xeCJK_boundary_hmode_transparent_begin:
\__xeCJK_boundary_hmode_transparent_end:
\__xeCJK_boundary_post_transparent:
\__xeCJK_boundary_post_transparent_relocate:
\__xeCJK_boundary_post_transparent_relocate_glue:
\__xeCJK_boundary_box_begin:
\__xeCJK_boundary_box_open:N
\__xeCJK_boundary_box_end:n
\__xeCJK_boundary_box_finish:nn
\__xeCJK_boundary_if_capture_box_sized:
\__xeCJK_boundary_if_capture_box_visible:
\__xeCJK_boundary_box_set_last_from_node:

```

表, 不会把离线 `\setbox` 中的公式直接认作外层可见输出。显式公式入口本身只处理当前列表, 不把事情广播到外层 `capture`, 因此不会改变裸公式和离线测量盒子的节点列表。

若公式后还有参数内尾随源码空格, 框架保留这枚空格在命令、框线或装饰范围内的原位置。普通 `stream` 在外层重放 `math-space marker`, 并随 `marker` 保存这枚空格的伸缩量; `box`、`wrapped-box` 和 `stream-ulem` 则重放 `math-space-frozen`, 表示空格已被冻结在内部盒子中。右侧为 `CJK` 且 `xCJKecglue` 启用时, 前者从补偿量中扣除实际仍参与外层断行的伸缩量; 后者完整保留 `\CJKecglue` 的伸缩量, 并禁止用负的自然宽度把后续文字拉入框线或装饰范围。命令外紧接的另一枚源码空格按连续源码空格处理, 显式 `glue` 则仍保留。`ulem` 会先把源码空格改造成 `leaders`, 所以它的参数适配器在交出尾随空格前确认公式节点, 再把空格交还 `ulem` 排版。若 `transparent` 或 `post-transparent` 命令把 `special`、零尺寸盒子等不可见节点插到普通 `stream` 的真实空格与 `marker` 之间, 直接公式的恢复链也会被同一节点截断; 框架因此让 `marker` 过期, 不在节点之后单独增加补偿 `glue`; 若 `marker` 与节点之间还有显式 `glue`, 也把它恢复到节点之前的原位置。宽、高、深均为零的盒子 (`\null`、空 `\mbox`)、只用于留白的盒子 (空 `\makebox`、`strut`、零高 `rule`), 以及末尾放着另一个已经排好的盒子或填充 `glue` 的命令 (例如 `\makebox` 放入 `\box`, `capture` 无法知道其中的字符类别), 都按“没有观察到可见输出”处理, 并恢复进入命令前的状态。源码空格、`\CJKecglue`、`\CJKglue` 及 `CJKspace`、`xCJKecglue` 状态都在入口字体和分组上下文中缓存, 避免命令内部字体或选项切换改变恢复值。`\sbox` 这类只构造离线盒子而不向当前列表输出的路径会暂停观察, 并按嵌套层保存、恢复外层 `marker` 与 `pending` 状态, 防止测量用 `scratch box` 或其中的颜色切换污染外层 `capture` 和基础恢复链。`ulem` 这类会把正文拆进固定宽度盒子的子系统使用专用 `stream kind`; 框架仍统一决定应发哪一种左侧 `glue`, 但把数值交回子系统自己的外层 `glue` 通道, 避免弹性分量被困在内部固定宽度盒子中。

额外 `hbox` 只用于 `\color@b@x` 这类本来会输出多个节点的注册命令; 普通盒子命令取出并放回原 `hbox`, 不增加节点层级。每个同步嵌套层使用独立 `register`, 避免外层盒子赋值尚未结束时被内层 `capture` 覆写; 常用的前三层在加载时预分配, 更多层则随首次达到的 `depth` 惰性创建。若 `wrapped-box` 没有观察到可见字符, 结束路径会 `unpack` 该层透明外层盒子。每层的 `token-list` 状态同样惰性分配, 使注册命令可以任意嵌套而不会静默退回未修复行为; 首类别会写入所有尚未取得首字符的外层 `capture`。`box` 结束时读取自身节点列表末尾由框架写入的 `marker`, 并据此更新末类别; 嵌套盒子的推断结果便能逐层传到最外面。`stream` 结束时也以外层节点列表末尾的 `marker` 为准; `annotation/math` 等没有末尾 `marker` 的流仍使用运行时观察结果。每层字段在 `\__xeCJK_boundary_capture_begin:` 中按同一 `depth` 连续完成 `clear`、求值与写入。当前不把 `flag` 序列化单独抽成 `helper`: 它仍需接收动态 `depth` 和不同字段名, 只会把可见的状态初始化拆散; 若以后继续增加每层字段, 再把 `allocate`、`initialize`、`restore` 一并评估为 `record/helper` 抽象。

实验性用户注册接口先把控制序列名、策略和类别模式写入独立属性表, 不立即修改命令。通用命令 `hook` 记录在注册表中; 必须直接重定义扫描器或内部排版入口的专用适配器另写入保留表。用户声明在保存时和导言区末尾应用时都查询这两张表, 不能把通用 `hook` 叠加到已有专用适配器外层。随后再调用同一组五种注册函数; LaTeX 在普通 `\AtBeginDocument` 代码执行完以后安装命令 `hook`。用户接口因此不复制恢复算法, 也不改变内建注册的优先级。正文开始前再检查成功登记的控制序列是否存在, 避免 LaTeX 对未定义命令的延迟 `hook` 静默失效。

```

1310 \int_new:N \g__xeCJK_boundary_capture_depth_int
1311 \int_new:N \g__xeCJK_boundary_suspend_depth_int
1312 \seq_new:N \g__xeCJK_boundary_active_seq
1313 \prop_new:N \g__xeCJK_boundary_registered_prop
1314 \prop_new:N \g__xeCJK_boundary_reserved_prop
1315 \prop_new:N \g__xeCJK_boundary_user_registered_prop
1316 \seq_new:N \g__xeCJK_boundary_user_applied_seq
1317 \bool_new:N \g__xeCJK_boundary_user_register_closed_bool
1318 \bool_new:N \l__xeCJK_boundary_user_invalid_bool
1319 \bool_new:N \l__xeCJK_boundary_user_mode_given_bool

```

```

1320 \tl_new:N \l__xeCJK_boundary_user_name_tl
1321 \tl_new:N \l__xeCJK_boundary_user_strategy_tl
1322 \tl_new:N \l__xeCJK_boundary_user_mode_tl
1323 \tl_new:N \l__xeCJK_boundary_user_value_tl
1324 \seq_new:N \l__xeCJK_boundary_user_value_seq
1325 \box_new:c { g__xeCJK_boundary_capture_1_box }
1326 \box_new:c { g__xeCJK_boundary_capture_2_box }
1327 \box_new:c { g__xeCJK_boundary_capture_3_box }
1328 \box_new:N \l__xeCJK_boundary_post_box
1329 \tl_new:N \l__xeCJK_boundary_node_tl
1330 \tl_new:N \l__xeCJK_boundary_first_tl
1331 \tl_new:N \l__xeCJK_boundary_last_tl
1332 \tl_new:N \l__xeCJK_boundary_active_tl
1333
1334 \cs_new_protected:Npn \__xeCJK_boundary_capture_allocate:n #1
1335 {
1336   \box_if_exist:cF { g__xeCJK_boundary_capture_#1_box }
1337   { \box_new:c { g__xeCJK_boundary_capture_#1_box } }
1338   \tl_if_exist:cF { g__xeCJK_boundary_capture_#1_before_tl }
1339   {
1340     \tl_new:c { g__xeCJK_boundary_capture_#1_before_tl }
1341     \tl_new:c { g__xeCJK_boundary_capture_#1_before_math_space_skip_tl }
1342     \tl_new:c { g__xeCJK_boundary_capture_#1_first_tl }
1343     \tl_new:c { g__xeCJK_boundary_capture_#1_last_tl }
1344     \tl_new:c { g__xeCJK_boundary_capture_#1_kind_tl }
1345     \tl_new:c { g__xeCJK_boundary_capture_#1_space_skip_tl }
1346     \tl_new:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl }
1347     \tl_new:c { g__xeCJK_boundary_capture_#1_ccglue_skip_tl }
1348     \tl_new:c { g__xeCJK_boundary_capture_#1_space_flag_tl }
1349     \tl_new:c { g__xeCJK_boundary_capture_#1_reserve_space_flag_tl }
1350     \tl_new:c { g__xeCJK_boundary_capture_#1_xecglue_flag_tl }
1351     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_flag_tl }
1352     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_space_flag_tl }
1353     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_space_skip_tl }
1354   }
1355 }
1356 \__xeCJK_boundary_capture_allocate:n { 1 }
1357 \__xeCJK_boundary_capture_allocate:n { 2 }
1358 \__xeCJK_boundary_capture_allocate:n { 3 }
1359
1360 \cs_new_protected:Npn \__xeCJK_boundary_capture_class:n #1
1361 {
1362   \int_compare:nNtT
1363   { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1364   {
1365     \int_compare:nNtT
1366     { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1367     {
1368       \int_step_inline:nn { \g__xeCJK_boundary_capture_depth_int }
1369       {
1370         \tl_gset:cn { g__xeCJK_boundary_capture_##1_last_tl } {#1}
1371         \tl_if_empty:cT { g__xeCJK_boundary_capture_##1_first_tl }
1372         {
1373           \tl_gset:cn
1374           { g__xeCJK_boundary_capture_##1_first_tl } {#1}
1375           \tl_if_eq:cnTF
1376           { g__xeCJK_boundary_capture_##1_kind_tl } { stream }
1377           { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1378           {
1379             \tl_if_eq:cnT
1380             { g__xeCJK_boundary_capture_##1_kind_tl } { stream-ulem }
1381             { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1382           }
1383         }
1384       }
1385     }
1386   }

```

```

1387 }
1388
1389 \cs_new_protected:Npn \__xeCJK_boundary_math_begin:
1390 {
1391   \mode_if_horizontal:T
1392   {
1393     \scan_stop:
1394     \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1395     \str_case:Vn \l__xeCJK_boundary_node_tl
1396     {
1397       { CJK }
1398       { \CJKe glue }
1399       { CJK-space }
1400       { \xeCJK_space_or_xec glue: }
1401       { CJK-widow }
1402       { \xeCJK_widow_penalty: \CJKe glue }
1403       { default-space }
1404       { \xeCJK_space_glue: }
1405     }
1406   }
1407 }
1408
1409 \cs_new_protected:Npn \__xeCJK_boundary_math_begin_visible:
1410 {
1411   \__xeCJK_boundary_capture_class:n { math }
1412   \__xeCJK_boundary_math_begin:
1413 }
1414
1415 \tl_new:N \l__xeCJK_boundary_math_tl
1416 \tl_new:N \l__xeCJK_boundary_math_type_tl
1417 \skip_new:N \l__xeCJK_boundary_math_skip
1418 \skip_new:N \l__xeCJK_boundary_math_actual_skip
1419 \skip_new:N \l__xeCJK_boundary_math_ec glue_skip
1420 \skip_new:N \l__xeCJK_boundary_math_delta_skip
1421 \dim_new:N \l__xeCJK_boundary_math_space_dim
1422 \dim_new:N \l__xeCJK_boundary_math_ec glue_dim
1423 \dim_new:N \l__xeCJK_boundary_math_stretch_dim
1424 \dim_new:N \l__xeCJK_boundary_math_shrink_dim
1425
1426 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_head:n #1
1427 { T , F , TF }
1428 {
1429   \tl_set:Nx \l__xeCJK_boundary_math_tl
1430   { \tl_trim_spaces:n {#1} }
1431   \tl_if_empty:NTF \l__xeCJK_boundary_math_tl
1432   { \prg_return_false: }
1433   {
1434     \exp_args:NV \tl_if_head_is_group:nTF
1435     \l__xeCJK_boundary_math_tl
1436     {
1437       \exp_args:Nx \__xeCJK_boundary_if_math_head:nTF
1438       { \tl_head:V \l__xeCJK_boundary_math_tl }
1439       { \prg_return_true: }
1440       { \prg_return_false: }
1441     }
1442     {
1443       \exp_args:NV \tl_if_head_eq_meaning:nNTF
1444       \l__xeCJK_boundary_math_tl $
1445       { \prg_return_true: }
1446       {
1447         \exp_args:NV \tl_if_head_eq_meaning:nNTF
1448         \l__xeCJK_boundary_math_tl \ (
1449         { \prg_return_true: }
1450         {
1451           \exp_args:NV \tl_if_head_eq_meaning:nNTF
1452           \l__xeCJK_boundary_math_tl \ensuremath
1453           { \prg_return_true: }

```

```

1454         { \prg_return_false: }
1455     }
1456 }
1457 }
1458 }
1459 }
1460
1461 \prg_new_protected_conditional:Npnn \_xeCJK_boundary_if_math_tail:n #1
1462 { T , F , TF }
1463 {
1464     \tl_set:Nc \l__xeCJK_boundary_math_tl
1465         { \tl_trim_spaces:n {#1} }
1466     \tl_reverse:N \l__xeCJK_boundary_math_tl
1467     \exp_args:NV \_xeCJK_boundary_if_math_tail_reversed:nTF
1468         \l__xeCJK_boundary_math_tl
1469     { \prg_return_true: }
1470     { \prg_return_false: }
1471 }
1472
1473 % \cs{tl_reverse:N} 只反转顶层项目，保留花括号组内的次序。末项目为组时，
1474 % 先识别 \tn{ensuremath} 的参数，否则递归检查组内末尾。所有语法判断都只
1475 % 产生候选：未知宏也可能把末尾的分组、|$| 或 \tn{)} 当作参数消费，
1476 % \cs{__xeCJK_boundary_math_end:n} 还要用当前列表的实际 math 节点或 marker
1477 % 确认，才会发布末类别。
1478 \prg_new_protected_conditional:Npnn
1479     \_xeCJK_boundary_if_math_tail_reversed:n #1 { T , F , TF }
1480 {
1481     \tl_if_empty:nTF {#1}
1482     { \prg_return_false: }
1483     {
1484         \tl_if_head_eq_meaning:nNTF {#1} $
1485         { \prg_return_true: }
1486         {
1487             \tl_if_head_eq_meaning:nNTF {#1} \
1488             { \prg_return_true: }
1489             {
1490                 \tl_if_head_is_group:nTF {#1}
1491                 {
1492                     \tl_set:Nc \l__xeCJK_boundary_math_tl
1493                         { \tl_trim_left_spaces:e { \tl_tail:n {#1} } }
1494                     \exp_args:NV \tl_if_head_eq_meaning:nNTF
1495                         \l__xeCJK_boundary_math_tl \ensuremath
1496                     { \prg_return_true: }
1497                     {
1498                         \exp_args:Nc \_xeCJK_boundary_if_math_tail:nTF
1499                             { \tl_head:n {#1} }
1500                         { \prg_return_true: }
1501                         { \prg_return_false: }
1502                     }
1503                 }
1504                 { \prg_return_false: }
1505             }
1506         }
1507     }
1508 }
1509
1510 \prg_new_protected_conditional:Npnn \_xeCJK_boundary_if_math_edge:n #1
1511 { T , F , TF }
1512 {
1513     \_xeCJK_boundary_if_math_head:nTF {#1}
1514     { \prg_return_true: }
1515     {
1516         \_xeCJK_boundary_if_math_tail:nTF {#1}
1517         { \prg_return_true: }
1518         { \prg_return_false: }
1519     }
1520 }

```

```

1521
1522 \cs_new_protected:Npn \__xeCJK_boundary_math_begin:n #1
1523 {
1524   \__xeCJK_boundary_if_math_head:nT {#1}
1525   { \__xeCJK_boundary_math_begin_visible: }
1526 }
1527
1528 \cs_new_protected:Npn \__xeCJK_boundary_math_end:n #1
1529 {
1530   \__xeCJK_boundary_if_math_tail:nT {#1}
1531   {
1532     \__xeCJK_boundary_if_math_tail_space:nTF {#1}
1533     { \__xeCJK_boundary_math_end_confirm_space: }
1534     { \__xeCJK_boundary_math_end_confirm_node: }
1535   }
1536 }
1537
1538 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_tail_space:n #1
1539 { T , F , TF }
1540 {
1541   \tl_set:Nn \l__xeCJK_boundary_math_tl {#1}
1542   \tl_set:Ne \l__xeCJK_boundary_node_tl
1543   { \tl_trim_right_spaces:n {#1} }
1544   \tl_if_eq:NNTF \l__xeCJK_boundary_math_tl \l__xeCJK_boundary_node_tl
1545   { \prg_return_false: }
1546   { \prg_return_true: }
1547 }
1548
1549 \prg_new_conditional:Npnn \__xeCJK_boundary_if_last_math_node: { T , F , TF }
1550 {
1551   \__xeCJK_if_last_math:TF
1552   { \prg_return_true: }
1553   {
1554     \xeCJK_if_last_node:nTF { math }
1555     { \prg_return_true: }
1556     { \prg_return_false: }
1557   }
1558 }
1559
1560 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_node:
1561 {
1562   \__xeCJK_boundary_if_last_math_node:T
1563   { \__xeCJK_boundary_math_end_confirm: }
1564 }
1565
1566 % 参数末尾的字面空格通常会成为一枚普通词间 glue，遮住它前面的公式节点。
1567 % 这里只允许越过语法扫描已经确认的尾随空格，并再次检查节点确为当前字体的
1568 % 词间空格；未知宏留下的其他 glue 不会进入这条路径。确认后把 glue 原样
1569 % 放回命令内部，并记录“公式后有参数内空格”的末类别，供 capture 结束时
1570 % 按普通 stream 或冻结空格分别处理。
1571 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_space:
1572 {
1573   \__xeCJK_if_last_glue:T
1574   {
1575     \skip_set_eq:NN \l__xeCJK_boundary_math_skip \tex_lastskip:D
1576     \__xeCJK_skip_if_interword:NT \l__xeCJK_boundary_math_skip
1577     {
1578       \tex_unskip:D
1579       \__xeCJK_boundary_if_last_math_node:T
1580       {
1581         \__xeCJK_boundary_math_end_confirm_space_visible:
1582       }
1583       \skip_horizontal:N \l__xeCJK_boundary_math_skip
1584     }
1585   }
1586 }
1587

```

```

1588 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_space_visible:
1589 {
1590   \__xeCJK_boundary_math_end_confirm:
1591   \tl_gset:cn
1592   {
1593     g__xeCJK_boundary_capture_
1594     \int_use:N \g__xeCJK_boundary_capture_depth_int
1595     _math_tail_space_flag_tl
1596   }
1597   { true }
1598   \tl_if_eq:cnT
1599   {
1600     g__xeCJK_boundary_capture_
1601     \int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl
1602   }
1603   { stream }
1604   {
1605     \tl_gset:ce
1606     {
1607       g__xeCJK_boundary_capture_
1608       \int_use:N \g__xeCJK_boundary_capture_depth_int
1609       _math_tail_space_skip_tl
1610     }
1611     { \skip_use:N \l__xeCJK_boundary_math_skip }
1612   }
1613 }
1614
1615 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm:
1616 {
1617   \int_compare:nNnT
1618   { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1619   {
1620     \int_compare:nNnT
1621     { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1622     {
1623       \tl_gset:cn
1624       {
1625         g__xeCJK_boundary_capture_
1626         \int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl
1627       }
1628       { math }
1629       \tl_gset:cn
1630       {
1631         g__xeCJK_boundary_capture_
1632         \int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
1633       }
1634       { true }
1635       \tl_gset:cn
1636       {
1637         g__xeCJK_boundary_capture_
1638         \int_use:N \g__xeCJK_boundary_capture_depth_int
1639         _math_tail_space_flag_tl
1640       }
1641       { false }
1642     }
1643   }
1644 }
1645
1646 \cs_new_protected:Npn \__xeCJK_boundary_capture_report_first:n #1
1647 {
1648   \int_compare:nNnT
1649   { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1650   {
1651     \int_compare:nNnT
1652     { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1653     {
1654       \int_step_inline:nn { \g__xeCJK_boundary_capture_depth_int }

```

```

1655         {
1656             \tl_if_empty:cT { g__xeCJK_boundary_capture_##1_first_tl }
1657             {
1658                 \tl_gset:cn
1659                 { g__xeCJK_boundary_capture_##1_first_tl } { #1 }
1660                 \tl_if_eq:cnTF
1661                 { g__xeCJK_boundary_capture_##1_kind_tl } { stream }
1662                 { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1663                 {
1664                     \tl_if_eq:cnT
1665                     { g__xeCJK_boundary_capture_##1_kind_tl } { stream-ulem }
1666                     { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1667                 }
1668             }
1669         }
1670     }
1671 }
1672 }
1673
1674 \cs_new_protected:Npn \__xeCJK_boundary_capture_emit_left:nn #1#2
1675 {
1676     \tl_set:Ne \l__xeCJK_boundary_node_tl
1677     { \tl_use:c { g__xeCJK_boundary_capture_#1_before_tl } }
1678     \__xeCJK_boundary_emit_left:nVn
1679     { #1 } \l__xeCJK_boundary_node_tl { #2 }
1680     \tl_gset:cn
1681     { g__xeCJK_boundary_capture_#1_space_flag_tl } { false }
1682 }
1683
1684 \cs_new_protected:Npn \__xeCJK_boundary_capture_suspend:
1685 {
1686     \int_gincr:N \g__xeCJK_boundary_suspend_depth_int
1687     \tl_if_exist:cF
1688     { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1689     {
1690         \tl_new:c
1691         { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1692         \tl_new:c
1693         { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _pending_tl }
1694     }
1695     \exp_args:Nc \tl_gset_eq:NN
1696     { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1697     \g__xeCJK_last_node_tl
1698     \tl_gset:ce
1699     { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _pending_tl }
1700     { \bool_if:NTF \g__xeCJK_glue_check_pending_bool { true } { false } }
1701 }
1702
1703 \cs_new_protected:Npn \__xeCJK_boundary_capture_resume:
1704 {
1705     \tl_gset:Ne \g__xeCJK_last_node_tl
1706     {
1707         \tl_use:c
1708         { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1709     }
1710     \tl_if_eq:cnTF
1711     { g__xeCJK_boundary_suspend_\int_use:N \g__xeCJK_boundary_suspend_depth_int _pending_tl }
1712     { true }
1713     { \bool_gset_true:N \g__xeCJK_glue_check_pending_bool }
1714     { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
1715     \int_gdecr:N \g__xeCJK_boundary_suspend_depth_int
1716 }
1717
1718 \cs_new_protected:Npn \__xeCJK_boundary_inline_box_begin:
1719 {
1720     \int_compare:nNnTF
1721     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int

```

```

1722     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1723     {
1724       \mode_if_math:TF
1725       { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1726       {
1727         \mode_leave_vertical:
1728         \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1729         \__xeCJK_boundary_box_begin:
1730       }
1731     }
1732   }
1733
1734   \cs_new_protected:Npn \__xeCJK_boundary_inline_box_end:n #1
1735   {
1736     \seq_gpop:NN
1737     \g__xeCJK_boundary_active_seq
1738     \l__xeCJK_boundary_active_tl
1739     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1740     { \__xeCJK_boundary_box_end:n {#1} }
1741   }
1742
1743   \cs_new_protected:Npn \__xeCJK_boundary_inline_last_box_begin:
1744   {
1745     \int_compare:nNnTF
1746     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1747     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1748     {
1749       \mode_if_math:TF
1750       { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1751       {
1752         \mode_leave_vertical:
1753         \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1754         \__xeCJK_boundary_capture_begin:
1755         \tl_gset:cn
1756         { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
1757         { box }
1758       }
1759     }
1760   }
1761
1762   \cs_new_protected:Npn \__xeCJK_boundary_inline_last_box_end:n #1
1763   {
1764     \seq_gpop:NN
1765     \g__xeCJK_boundary_active_seq
1766     \l__xeCJK_boundary_active_tl
1767     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1768     { \__xeCJK_boundary_last_box_end:n {#1} }
1769   }
1770
1771   \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_begin:n #1
1772   { \__xeCJK_boundary_inline_stream_begin:nn {#1} { stream } }
1773
1774   \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_begin:nn #1#2
1775   {
1776     \int_compare:nNnTF
1777     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1778     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1779     {
1780       \mode_if_math:TF
1781       { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1782       {
1783         \mode_leave_vertical:
1784         \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1785         \__xeCJK_boundary_capture_begin:
1786         \tl_gset:cn
1787         { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
1788         {#2}

```

```

1789         \str_case:nn {#1}
1790         {
1791             { default }          { \__xeCJK_boundary_capture_class:n { default } }
1792             { first-default } { \__xeCJK_boundary_capture_class:n { default } }
1793         }
1794     }
1795 }
1796 }
1797
1798 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_end:n #1
1799 {
1800     \seq_gpop:Nn
1801     \g__xeCJK_boundary_active_seq
1802     \l__xeCJK_boundary_active_tl
1803     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1804     {
1805         \scan_stop:
1806         \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1807         \tl_set:Nc \l__xeCJK_boundary_first_tl
1808         {
1809             \tl_use:c
1810             { g__xeCJK_boundary_capture\_int\_use:N \g__xeCJK_boundary_capture_depth\_int\_first\_tl }
1811         }
1812         \tl_set:Nc \l__xeCJK_boundary_last_tl
1813         {
1814             \tl_use:c
1815             { g__xeCJK_boundary_capture\_int\_use:N \g__xeCJK_boundary_capture_depth\_int\_last\_tl }
1816         }
1817         % 流式命令若在最外层留下 xeCJK marker，就以该 marker 记录的
1818         % 类别更新末类别。这适用于内部使用不可见定界字符、最后又把正文末尾
1819         % marker 移到外层列表的命令；没有末尾 marker 的 annotation/math
1820         % 仍使用 capture 观察值。
1821         \exp_args:NV
1822         \__xeCJK_boundary_set_last_from_node:n \l__xeCJK_boundary_node_tl
1823         \__xeCJK_boundary_set_math_tail_from_capture:
1824         % |default| 的开始 hook 已经固定首类别；这里仅覆写末类别即可让
1825         % stream 两端都为 Default，同时不丢弃其他模式实际观察到的首类别。
1826         \str_if_eq:nnT {#1} { default }
1827         { \tl_set:Nn \l__xeCJK_boundary_last_tl {#1} }
1828         \tl_gclear:N \g__xeCJK_last_node_tl
1829         \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1830         \tl_if_empty:NNT \l__xeCJK_boundary_first_tl
1831         {
1832             \__xeCJK_boundary_replay_before:
1833         }
1834         {
1835             \tl_if_empty:NT \l__xeCJK_boundary_last_tl
1836             { \tl_set_eq:NN \l__xeCJK_boundary_last_tl \l__xeCJK_boundary_first_tl }
1837             \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_last_tl
1838         }
1839         \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
1840     }
1841 }
1842
1843 \cs_new_protected:Npn \__xeCJK_boundary_hmode_transparent_begin:
1844 {
1845     \int_compare:nNnTF
1846     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1847     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1848     {
1849         \mode_if_horizontal:TF
1850         {
1851             \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1852             \__xeCJK_boundary_capture_begin:
1853             \tl_gset:cn
1854             { g__xeCJK_boundary_capture\_int\_use:N \g__xeCJK_boundary_capture_depth\_int\_kind\_tl }
1855             { transparent }

```

```

1856     }
1857     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1858   }
1859 }
1860
1861 \cs_new_protected:Npn \__xeCJK_boundary_hmode_transparent_end:
1862 {
1863   \seq_gpop:NN
1864   \g__xeCJK_boundary_active_seq
1865   \l__xeCJK_boundary_active_tl
1866   \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1867   {
1868     \__xeCJK_boundary_replay_before:
1869     \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
1870   }
1871 }
1872
1873 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent:
1874 {
1875   \mode_if_horizontal:T
1876   {
1877     \tl_if_empty:NF \g__xeCJK_last_node_tl
1878     {
1879       \__xeCJK_if_last_hlist:TF
1880       {
1881         \box_set_to_last:N \l__xeCJK_boundary_post_box
1882         \dim_compare:nNnTF
1883         {
1884           \box_wd:N \l__xeCJK_boundary_post_box
1885           + \box_ht:N \l__xeCJK_boundary_post_box
1886           + \box_dp:N \l__xeCJK_boundary_post_box
1887         }
1888         = { \c_zero_dim }
1889         { \__xeCJK_boundary_post_transparent_relocate: }
1890         { \box_use_drop:N \l__xeCJK_boundary_post_box }
1891       }
1892       {
1893         \xeCJK_if_last_node:TF
1894         {
1895           \exp_args:NV
1896           \__xeCJK_boundary_replay_node:n \g__xeCJK_last_node_tl
1897         }
1898         { }
1899       }
1900     }
1901   }
1902 }
1903
1904 % 零尺寸盒子前可能不只有 marker，还可能有一枚等待 Boundary 恢复链检查的
1905 % glue。第一层先处理盒子直接盖住 marker 的情况；未命中时才暂存至多一枚
1906 % 末尾 glue，再按节点证据检查其下方的 marker。命中后把“marker + glue”整体
1907 % 移到盒子后面，使后续转换看到与没有该盒子时相同的列表末尾。未命中 marker
1908 % 时按原顺序还原，不能把任意 glue 搬过 hbox。普通 stream 的  $space$ 
1909 % 还有一枚真实空格留在 marker 前；零尺寸盒子已经把二者隔开时，应与含同一
1910 % 盒子的直接公式一样让 marker 过期，不能只把补偿状态搬到盒子后面。
1911 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent_relocate:
1912 {
1913   \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1914   \tl_if_empty:NTF \l__xeCJK_boundary_node_tl
1915   {
1916     \__xeCJK_if_last_glue:TF
1917     { \__xeCJK_boundary_post_transparent_relocate_glue: }
1918     { \box_use_drop:N \l__xeCJK_boundary_post_box }
1919   }
1920   {
1921     % 盒子直接盖住 marker，中间没有 glue。
1922     \box_use_drop:N \l__xeCJK_boundary_post_box

```

```

1923     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
1924     { \__xeCJK_boundary_math_space_expire: }
1925     {
1926         \exp_args:NV
1927         \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
1928     }
1929 }
1930 }
1931
1932 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent_relocate_glue:
1933 {
1934     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1935     \tex_unskip:D
1936     \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1937     \tl_if_empty:NTF \l__xeCJK_boundary_node_tl
1938     {
1939         \skip_horizontal:N \l__xeCJK_last_skip
1940         \box_use_drop:N \l__xeCJK_boundary_post_box
1941     }
1942     {
1943         \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
1944         {
1945             % |math-space| 过期时不再移动后缀；先恢复候选 glue，再放回
1946             % 零尺寸盒子，保持与含同一盒子的直接公式相同的节点顺序。
1947             \skip_horizontal:N \l__xeCJK_last_skip
1948             \box_use_drop:N \l__xeCJK_boundary_post_box
1949             \__xeCJK_boundary_math_space_expire:
1950         }
1951         {
1952             \box_use_drop:N \l__xeCJK_boundary_post_box
1953             \exp_args:NV
1954             \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
1955             \skip_horizontal:N \l__xeCJK_last_skip
1956         }
1957     }
1958 }
1959
1960 \cs_new_protected:Npn \__xeCJK_boundary_capture_begin:
1961 {
1962     \int_gincr:N \g__xeCJK_boundary_capture_depth_int
1963     \int_compare:nNnT
1964     { \g__xeCJK_boundary_capture_depth_int } > { 2 }
1965     {
1966         \exp_args:Ne \__xeCJK_boundary_capture_allocate:n
1967         { \int_use:N \g__xeCJK_boundary_capture_depth_int }
1968     }
1969     \tl_gclear:c
1970     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl }
1971     \tl_gclear:c
1972     {
1973         g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int
1974         _before_math_space_skip_tl
1975     }
1976     \tl_gclear:c
1977     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _first_tl }
1978     \tl_gclear:c
1979     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl }
1980     \tl_gclear:c
1981     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
1982     % 先把用户定义的 glue 实际排入临时盒并读取为 skip；后面的 e-type 只把
1983     % 已求值的 skip 数值序列化到本层 tl，不展开原始 \CJKglue/\CJKecglue 定义。
1984     \xeCJK_glue_to_skip:nN
1985     { \tex_kern:D \c_zero_dim \xeCJK_space_glue: }
1986     \l__xeCJK_tmp_skip
1987     \tl_gset:ce
1988     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_skip_tl }
1989     { \skip_use:N \l__xeCJK_tmp_skip }

```

```

1990 \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_tmp_skip
1991 \tl_gset:ce
1992 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _ecglue_skip_tl
1993   { \skip_use:N \l__xeCJK_tmp_skip }
1994 \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_tmp_skip
1995 \tl_gset:ce
1996 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _ccglue_skip_tl
1997   { \skip_use:N \l__xeCJK_tmp_skip }
1998 \tl_gset:cn
1999 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_tl }
2000 { false }
2001 \tl_gset:ce
2002 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _reserve_space_f
2003   { \bool_if:NTF \l__xeCJK_reserve_space_bool { true } { false } }
2004 \tl_gset:ce
2005 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _xecglue_flag_tl
2006   { \bool_if:NTF \l__xeCJK_xecglue_bool { true } { false } }
2007 \tl_gset:cn
2008 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag
2009   { false }
2010 \tl_gset:cn
2011 {
2012   g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int
2013   _math_tail_space_flag_tl
2014 }
2015 { false }
2016 \tl_gclear:c
2017 {
2018   g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int
2019   _math_tail_space_skip_tl
2020 }
2021 \__xeCJK_boundary_capture_space:
2022 \scan_stop:
2023 \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2024 \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { math-space }
2025 {
2026   \__xeCJK_if_last_glue:TF
2027   {
2028     \tl_gset:ce
2029     {
2030       g__xeCJK_boundary_capture_
2031       \int_use:N \g__xeCJK_boundary_capture_depth_int
2032       _before_math_space_skip_tl
2033     }
2034     { \skip_use:N \tex_lastskip:D }
2035   }
2036   {
2037     \tl_gset:ce
2038     {
2039       g__xeCJK_boundary_capture_
2040       \int_use:N \g__xeCJK_boundary_capture_depth_int
2041       _before_math_space_skip_tl
2042     }
2043     { \skip_use:N \l__xeCJK_boundary_math_actual_skip }
2044   }
2045 }
2046 \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { CJK-space }
2047 {
2048   \tl_gset:cn
2049   { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_t
2050     { true }
2051 }
2052 \tl_if_empty:NT \l__xeCJK_boundary_node_tl
2053 {
2054   \tl_if_eq:cnT
2055   { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_t
2056     { true }

```

```

2057     {
2058         \__xeCJK_boundary_restore_space:
2059         \tl_gset:cn
2060         { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag
2061           { false }
2062         }
2063     }
2064     \exp_args:Nc \tl_gset_eq:NN
2065     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl }
2066     \l__xeCJK_boundary_node_tl
2067     \tl_gclear:N \g__xeCJK_last_node_tl
2068     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2069 }
2070
2071 \cs_new_protected:Npn \__xeCJK_boundary_box_begin:
2072 {
2073     \__xeCJK_boundary_capture_begin:
2074     \tl_gset:cn
2075     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
2076     { box }
2077     \exp_args:Nc \__xeCJK_boundary_box_open:N
2078     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2079 }
2080
2081 \cs_new_protected:Npn \__xeCJK_boundary_box_open:N #1
2082 {
2083     \tex_global:D \tex_setbox:D #1
2084     \tex_hbox:D \c_group_begin_token
2085 }
2086
2087 \cs_new_protected:Npn \__xeCJK_boundary_capture_space:
2088 {
2089     \__xeCJK_if_last_glue:T
2090     {
2091         \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
2092         \__xeCJK_skip_if_interword:NT \l__xeCJK_last_skip
2093         {
2094             \tl_gset:ce
2095             { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_skip
2096               { \skip_use:N \l__xeCJK_last_skip }
2097             \tl_gset:cn
2098             { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag
2099               { true }
2100             \tex_unskip:D
2101           }
2102         }
2103     }
2104
2105 \cs_new_protected:Npn \__xeCJK_boundary_pop_node:N #1
2106 {
2107     \tl_clear:N #1
2108     \xeCJK_if_last_node:TF
2109     {
2110         \dim_case:nnF { \l__xeCJK_last_kern_dim }
2111         {
2112             { \__xeCJK_node:n { CJK } } { \tl_set:Nn #1 { CJK } }
2113             { \__xeCJK_node:n { CJK-space } } { \tl_set:Nn #1 { CJK-space } }
2114             { \__xeCJK_node:n { CJK-widow } } { \tl_set:Nn #1 { CJK-widow } }
2115             { \__xeCJK_node:n { default } } { \tl_set:Nn #1 { default } }
2116             { \__xeCJK_node:n { math } } { \tl_set:Nn #1 { math } }
2117             { \__xeCJK_node:n { math-space } }
2118             {
2119                 \tl_set:Nn #1 { math-space }
2120                 \__xeCJK_boundary_math_space_decode:
2121             }
2122             { \__xeCJK_node:n { math-space-frozen } }
2123             { \tl_set:Nn #1 { math-space-frozen } }

```

```

2124         { \_xeCJK_node:n { normalspace } }      { \tl_set:Nn #1 { normalspace } }
2125     }
2126     { \_xeCJK_make_node:N \l__xeCJK_last_kern_dim }
2127 }
2128 {
2129     \xeCJK_if_last_node:nT { default-space }
2130     {
2131         \xeCJK_remove_node:
2132         \tl_set:Nn #1 { default-space }
2133     }
2134 }
2135 }
2136
2137 \prg_new_conditional:Npnn \_xeCJK_boundary_if_last_math_space: { T , F , TF }
2138 {
2139     \xeCJK_if_last_node:nTF { math-space }
2140     { \prg_return_true: }
2141     {
2142         \xeCJK_if_last_node:nTF { math-space-frozen }
2143         { \prg_return_true: }
2144         { \prg_return_false: }
2145     }
2146 }
2147
2148 \cs_new_protected:Npn \_xeCJK_boundary_math_space_consume:
2149 {
2150     \xeCJK_if_last_node:nTF { math-space }
2151     {
2152         \xeCJK_remove_node:
2153         \_xeCJK_boundary_math_space_decode:
2154     }
2155     { \xeCJK_remove_node: }
2156     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2157 }
2158
2159 % |math-space| 后面的两对零净宽 kern 依次保存普通 stream 空格的伸长量
2160 % 和收缩量。marker 位于最外侧，移除 marker 后按相反次序取回两对 kern。
2161 % 这里只保存有限阶词间空格的弹性分量；字体的普通词间空格不会使用 fil 阶。
2162 \cs_new_protected:Npn \_xeCJK_boundary_math_space_set_actual:n #1
2163 {
2164     \skip_set:Nn \l__xeCJK_boundary_math_actual_skip {#1}
2165     \dim_set:Nn \l__xeCJK_boundary_math_stretch_dim
2166     { \tex_gluestretch:D \l__xeCJK_boundary_math_actual_skip }
2167     \dim_set:Nn \l__xeCJK_boundary_math_shrink_dim
2168     { \tex_glueshrink:D \l__xeCJK_boundary_math_actual_skip }
2169     \skip_set:Nn \l__xeCJK_boundary_math_actual_skip
2170     {
2171         \c_zero_dim
2172         plus \l__xeCJK_boundary_math_stretch_dim
2173         minus \l__xeCJK_boundary_math_shrink_dim
2174     }
2175 }
2176
2177 \cs_new_protected:Npn \_xeCJK_boundary_math_space_encode:
2178 {
2179     \dim_set:Nn \l__xeCJK_boundary_math_stretch_dim
2180     { \tex_gluestretch:D \l__xeCJK_boundary_math_actual_skip }
2181     \dim_set:Nn \l__xeCJK_boundary_math_shrink_dim
2182     { \tex_glueshrink:D \l__xeCJK_boundary_math_actual_skip }
2183     \_xeCJK_make_node:N \l__xeCJK_boundary_math_stretch_dim
2184     \_xeCJK_make_node:N \l__xeCJK_boundary_math_shrink_dim
2185 }
2186
2187 \cs_new_protected:Npn \_xeCJK_boundary_math_space_decode:
2188 {
2189     \dim_zero:N \l__xeCJK_boundary_math_stretch_dim
2190     \dim_zero:N \l__xeCJK_boundary_math_shrink_dim

```

```

2191 \xeCJK_if_last_node:TF
2192 {
2193   \dim_set_eq:NN
2194   \l__xeCJK_boundary_math_shrink_dim \l__xeCJK_last_kern_dim
2195   \xeCJK_if_last_node:TF
2196   {
2197     \dim_set_eq:NN
2198     \l__xeCJK_boundary_math_stretch_dim \l__xeCJK_last_kern_dim
2199   }
2200   { }
2201 }
2202 { }
2203 \skip_set:Nn \l__xeCJK_boundary_math_actual_skip
2204 {
2205   \c_zero_dim
2206   plus \l__xeCJK_boundary_math_stretch_dim
2207   minus \l__xeCJK_boundary_math_shrink_dim
2208 }
2209 }
2210
2211 % 普通 stream 的参数内空格仍在外层列表：自然差额按入口字体的词间空格
2212 % 计算，伸缩差额则扣除参数内实际空格的伸缩量。box、wrapped-box 和
2213 % stream-ulem 已把空格冻结在内部盒子中；它的伸缩分量不再参与外层断行，
2214 % 外层只能扣除词间空格的自然宽度，并完整保留 |CJKecglue| 自己的伸缩分量。
2215 % 若目标自然宽度更小，则把外层自然差额截为零，不能用负 glue 把后续文字
2216 % 拉回框线或装饰范围。命令内外字体不同时，内部空格的宽度差仍属于命令
2217 % 自身的固有宽度。
2218 \cs_new_protected:Npn \__xeCJK_boundary_math_space_set_delta:n #1
2219 {
2220   \str_if_eq:nnTF {#1} { math-space-frozen }
2221   {
2222     \dim_set:Nn \l__xeCJK_boundary_math_space_dim
2223     { \l__xeCJK_boundary_math_skip }
2224     \dim_set:Nn \l__xeCJK_boundary_math_ecglue_dim
2225     { \l__xeCJK_boundary_math_ecglue_skip }
2226     \dim_set:Nn \l__xeCJK_tmp_dim
2227     {
2228       \dim_max:nn
2229       {
2230         \l__xeCJK_boundary_math_ecglue_dim
2231         - \l__xeCJK_boundary_math_space_dim
2232       }
2233       { \c_zero_dim }
2234     }
2235     \skip_set:Nn \l__xeCJK_boundary_math_delta_skip
2236     {
2237       \l__xeCJK_boundary_math_ecglue_skip
2238       - \l__xeCJK_boundary_math_ecglue_dim
2239       + \l__xeCJK_tmp_dim
2240     }
2241   }
2242   {
2243     \dim_set:Nn \l__xeCJK_boundary_math_space_dim
2244     { \l__xeCJK_boundary_math_skip }
2245     \dim_set:Nn \l__xeCJK_boundary_math_ecglue_dim
2246     { \l__xeCJK_boundary_math_ecglue_skip }
2247     \dim_set:Nn \l__xeCJK_tmp_dim
2248     {
2249       \l__xeCJK_boundary_math_ecglue_dim
2250       - \l__xeCJK_boundary_math_space_dim
2251     }
2252     \skip_set:Nn \l__xeCJK_boundary_math_delta_skip
2253     {
2254       \l__xeCJK_boundary_math_ecglue_skip
2255       - \l__xeCJK_boundary_math_ecglue_dim
2256       + \l__xeCJK_tmp_dim
2257       - \l__xeCJK_boundary_math_actual_skip

```

```

2258     }
2259   }
2260 }
2261
2262 \cs_new_protected:Npn \__xeCJK_boundary_math_space_to_CJK:
2263 {
2264   \xeCJK_if_last_node:nTF { math-space-frozen }
2265   { \tl_set:Nn \l__xeCJK_boundary_math_type_tl { math-space-frozen } }
2266   { \tl_set:Nn \l__xeCJK_boundary_math_type_tl { math-space } }
2267   \__xeCJK_boundary_math_space_consume:
2268   \bool_if:NT \l__xeCJK_xecglue_bool
2269   {
2270     \skip_set:Nn \l__xeCJK_boundary_math_skip
2271     { \c_xeCJK_space_skip_tl }
2272     \skip_set_eq:NN
2273     \l__xeCJK_boundary_math_ecglue_skip \l__xeCJK_ecglue_skip
2274     \exp_args:NV \__xeCJK_boundary_math_space_set_delta:n
2275     \l__xeCJK_boundary_math_type_tl
2276     \skip_if_eq:nnF
2277     { \l__xeCJK_boundary_math_delta_skip } { \c_zero_skip }
2278     { \skip_horizontal:N \l__xeCJK_boundary_math_delta_skip }
2279   }
2280 }
2281
2282 \cs_new_protected:Npn \__xeCJK_boundary_box_end:n #1
2283 {
2284   \c_group_end_token
2285   \__xeCJK_boundary_box_finish:nn {#1} { wrapped }
2286 }
2287
2288 \cs_new_protected:Npn \__xeCJK_boundary_last_box_end:n #1
2289 {
2290   \__xeCJK_if_last_hlist:TF
2291   {
2292     \exp_args:Nc \box_set_to_last:N
2293     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2294     \__xeCJK_boundary_box_finish:nn {#1} { last }
2295   }
2296   {
2297     \tl_gclear:N \g__xeCJK_last_node_tl
2298     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2299     \__xeCJK_boundary_replay_before:
2300     \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
2301   }
2302 }
2303
2304 \int_new:N \g__xeCJK_boundary_probe_type_int
2305 \box_new:N \l__xeCJK_boundary_probe_box
2306 \tl_new:N \l__xeCJK_boundary_probe_name_tl
2307 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_capture_box_sized:
2308 { T , F , TF }
2309 {
2310   \tl_set:Ne \l__xeCJK_boundary_probe_name_tl
2311   {
2312     g__xeCJK_boundary_capture_
2313     \int_use:N \g__xeCJK_boundary_capture_depth_int _box
2314   }
2315   \bool_lazy_and:nnTF
2316   {
2317     ! \dim_compare_p:nNn
2318     { \box_wd:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2319   }
2320   {
2321     \bool_lazy_or_p:nn
2322     {
2323       ! \dim_compare_p:nNn
2324       { \box_ht:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim

```

```

2325     }
2326     {
2327         ! \dim_compare_p:nNn
2328         { \box_dp:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2329     }
2330 }
2331 { \prg_return_true: }
2332 { \prg_return_false: }
2333 }
2334 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_capture_box_visible:
2335 { T , F , TF }
2336 {
2337     \__xeCJK_boundary_if_capture_box_sized:TF
2338     {
2339         % 只复制节点，不重扫 token，无排版副作用。
2340         \hbox_set:Nn \l__xeCJK_boundary_probe_box
2341         {
2342             \hbox_unpack:c { \l__xeCJK_boundary_probe_name_tl }
2343             \int_gset:Nn \g__xeCJK_boundary_probe_type_int
2344             { \tex_lastnodetype:D }
2345         }
2346         \box_clear:N \l__xeCJK_boundary_probe_box
2347         % \lastnodetype 编码沿用 eTeX 手册的节点类型表 (0 传统 char、
2348         % 1 hlist、3 rule、9 whatsit、10 math、11 glue、12 kern)；
2349         % XeTeX 把 native 文本实现为 word 节点、同样报 9，但它必然已被
2350         % interchar 观察，不会进入这里。这里把 char、rule、math，以及
2351         % 未上报标点类字符后由标点机制添加的 kern 视为盒子直接排出的内容。
2352         \int_case:nnTF { \g__xeCJK_boundary_probe_type_int }
2353         {
2354             { 0 } { }
2355             { 3 } { }
2356             { 10 } { }
2357             { 12 } { }
2358         }
2359         { \prg_return_true: }
2360         { \prg_return_false: }
2361     }
2362     { \prg_return_false: }
2363 }
2364
2365 \tl_new:N \g__xeCJK_boundary_probe_last_tl
2366 \cs_new_protected:Npn \__xeCJK_boundary_box_set_last_from_node:
2367 {
2368     \tl_set:Ne \l__xeCJK_boundary_probe_name_tl
2369     {
2370         g__xeCJK_boundary_capture_
2371         \int_use:N \g__xeCJK_boundary_capture_depth_int _box
2372     }
2373     \tl_gclear:N \g__xeCJK_boundary_probe_last_tl
2374     \hbox_set:Nn \l__xeCJK_boundary_probe_box
2375     {
2376         \hbox_unpack:c { \l__xeCJK_boundary_probe_name_tl }
2377         \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2378         \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { math-space }
2379         { \tl_set:Nn \l__xeCJK_boundary_node_tl { math-space-frozen } }
2380         \tl_gset_eq:NN
2381         \g__xeCJK_boundary_probe_last_tl \l__xeCJK_boundary_node_tl
2382     }
2383     \box_clear:N \l__xeCJK_boundary_probe_box
2384     \exp_args:NV
2385     \__xeCJK_boundary_set_last_from_node:n \g__xeCJK_boundary_probe_last_tl
2386 }
2387
2388 \cs_new_protected:Npn \__xeCJK_boundary_box_finish:nn #1#2
2389 {
2390     \tl_set:Ne \l__xeCJK_boundary_first_tl
2391     {

```

```

2392     \tl_use:c
2393     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _first_tl }
2394   }
2395   \tl_set:Nn \l__xeCJK_boundary_last_tl
2396   {
2397     \tl_use:c
2398     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl }
2399   }
2400   \tl_gclear:N \g__xeCJK_last_node_tl
2401   \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2402   % 嵌套 box 结束时写入的 marker 会留在当前盒子的节点列表末尾。
2403   % 读取这个 marker 可以更新本层的末类别，使“前面已经观察到 CJK，末尾
2404   % 是推断为 Default 的内容”等情况逐层传到外面；不能把命令内部辅助盒子的
2405   % 推断直接写进所有外层 capture。
2406   % 即使 capture 已记录末类别也不能跳过这一步：内部不可见的定界字符可能
2407   % 使观察值不准，此时应以节点列表末尾的 marker 为准。
2408   \__xeCJK_boundary_box_set_last_from_node:
2409   \__xeCJK_boundary_set_math_tail_from_capture:
2410   % 盒子捕获没有独立的 stream 入口，故 |default| 在这里同时覆写实际观察到
2411   % 的首尾类别；stream 路径则由开始 hook 固定首类别、结束 hook 固定
2412   % 末类别，两条路径的公开语义相同。
2413   \str_if_eq:nnT {#1} { default }
2414   {
2415     \tl_set:Nn \l__xeCJK_boundary_first_tl { default }
2416     \tl_set:Nn \l__xeCJK_boundary_last_tl { default }
2417   }
2418   % rule 和尚未由公式观察器记录的 math 不触发 interchar 转换。若盒子
2419   % 直接排出了可见内容
2420   % （宽度非零、高度或深度非零，且末节点为 char/rule/math/kern），
2421   % 却没有观察到任何类别，就按 Default 首尾重建边界。空 makebox、strut
2422   % 等只用于留白的盒子，以及放入另一个已经排好盒子的命令，仍按无可见
2423   % 输出处理（issue 998）。
2424   % 只把 default 报告给尚未取得首类别的外层 capture，使 \nfss@text
2425   % 这类嵌在 stream 内的盒子命令与裸调用等价；各层盒子结束时读取
2426   % 节点列表末尾由本层写入的 marker，并据此更新末类别。
2427   \tl_if_empty:NT \l__xeCJK_boundary_first_tl
2428   {
2429     \__xeCJK_boundary_if_capture_box_visible:T
2430     {
2431       \tl_set:Nn \l__xeCJK_boundary_first_tl { default }
2432       \tl_set:Nn \l__xeCJK_boundary_last_tl { default }
2433       \__xeCJK_boundary_capture_report_first:n { default }
2434     }
2435   }
2436   \tl_if_empty:NTF \l__xeCJK_boundary_first_tl
2437   { \__xeCJK_boundary_box_end_transparent:n {#2} }
2438   {
2439     \tl_set:Nn \l__xeCJK_boundary_node_tl
2440     {
2441       \tl_use:c
2442       { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl }
2443     }
2444     \exp_args:NVV \__xeCJK_boundary_emit_left:nn
2445     \l__xeCJK_boundary_node_tl
2446     \l__xeCJK_boundary_first_tl
2447     \box_use_drop:c
2448     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2449     \tl_if_empty:NT \l__xeCJK_boundary_last_tl
2450     { \tl_set_eq:NN \l__xeCJK_boundary_last_tl \l__xeCJK_boundary_first_tl }
2451     \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_last_tl
2452   }
2453   \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
2454 }
2455
2456 \cs_new_protected:Npn \__xeCJK_boundary_box_end_transparent:n #1
2457 {
2458   \str_if_eq:nnTF {#1} { wrapped }

```

```

2459 {
2460   \hbox_unpack_drop:c
2461   { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2462 }
2463 {
2464   \box_use_drop:c
2465   { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2466 }
2467 \__xeCJK_boundary_replay_before:
2468 }
2469
2470 \cs_new_protected:Npn \__xeCJK_boundary_replay_before:
2471 {
2472   \tl_set:Nc \l__xeCJK_boundary_node_tl
2473   {
2474     \tl_use:c
2475     {
2476       g__xeCJK_boundary_capture_
2477       \int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl
2478     }
2479   }
2480   \clist_if_in:nVTF
2481   { math-space , math-space-frozen } \l__xeCJK_boundary_node_tl
2482   {
2483     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
2484     { \__xeCJK_boundary_replay_before_math_space: }
2485     { \__xeCJK_boundary_replay_node:n { math-space-frozen } }
2486   }
2487   {
2488     \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
2489     \__xeCJK_boundary_restore_space:
2490   }
2491 }
2492
2493 % 普通 stream 的 |math-space| 只在真实参数空格仍紧邻列表末尾时重放。
2494 % transparent 命令若已经在二者之间写入 special、零尺寸盒子等节点，直接
2495 % 公式的源码空格同样不会越过该节点触发恢复；此时必须让 marker 过期，
2496 % 不能把补偿 glue 单独移到不可见节点之后。skip 比较把判断限制在原来那枚
2497 % 真实空格，不把 transparent 命令偶然写出的不同规格 glue 当作边界后缀。
2498 \cs_new_protected:Npn \__xeCJK_boundary_replay_before_math_space:
2499 {
2500   \__xeCJK_if_last_glue:T
2501   {
2502     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
2503     \skip_if_eq:nnT
2504     { \l__xeCJK_last_skip }
2505     {
2506       \tl_use:c
2507       {
2508         g__xeCJK_boundary_capture_
2509         \int_use:N \g__xeCJK_boundary_capture_depth_int
2510         _before_math_space_skip_tl
2511       }
2512     }
2513     {
2514       \exp_args:Nc \__xeCJK_boundary_math_space_set_actual:n
2515       {
2516         \tl_use:c
2517         {
2518           g__xeCJK_boundary_capture_
2519           \int_use:N \g__xeCJK_boundary_capture_depth_int
2520           _before_math_space_skip_tl
2521         }
2522       }
2523       \__xeCJK_boundary_replay_node:n { math-space }
2524     }
2525   }

```

```

2526 }
2527
2528 \cs_new_protected:Npn \__xeCJK_boundary_math_space_expire:
2529 {
2530   \tl_gclear:N \g__xeCJK_last_node_tl
2531   \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2532 }
2533
2534 \cs_new_protected:Npn \__xeCJK_boundary_emit_left:nn #1#2
2535 {
2536   \__xeCJK_boundary_emit_left:nnn
2537   { \int_use:N \g__xeCJK_boundary_capture_depth_int } {#1} {#2}
2538 }
2539
2540 \cs_new_protected:Npn \__xeCJK_boundary_emit_left:nnn #1#2#3
2541 {
2542   \clist_if_in:nnTF { math-space , math-space-frozen } {#2}
2543   { \__xeCJK_boundary_emit_left_math_space:nnn {#1} {#2} {#3} }
2544   { \__xeCJK_boundary_emit_left_normal:nnn {#1} {#2} {#3} }
2545 }
2546 \cs_new_protected:Npn \__xeCJK_boundary_emit_left_math_space:nnn #1#2#3
2547 {
2548   \str_if_eq:nnT {#3} { CJK }
2549   {
2550     \tl_if_eq:cnT
2551     { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2552     {
2553       \str_if_eq:nnT {#2} { math-space }
2554       {
2555         \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2556         {
2557           \tl_use:c
2558           { g__xeCJK_boundary_capture_#1_before_math_space_skip_tl }
2559         }
2560       }
2561       \skip_set:Nn \l__xeCJK_boundary_math_skip
2562       { \tl_use:c { g__xeCJK_boundary_capture_#1_space_skip_tl } }
2563       \skip_set:Nn \l__xeCJK_boundary_math_ecglue_skip
2564       { \tl_use:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl } }
2565       \__xeCJK_boundary_math_space_set_delta:n {#2}
2566       \skip_if_eq:nnF
2567       { \l__xeCJK_boundary_math_delta_skip } { \c_zero_skip }
2568       {
2569         \__xeCJK_boundary_use_glue:nn {#1}
2570         { \l__xeCJK_boundary_math_delta_skip }
2571       }
2572     }
2573   }
2574 }
2575 \cs_new_protected:Npn \__xeCJK_boundary_emit_left_normal:nnn #1#2#3
2576 {
2577   \clist_if_in:nnTF { CJK , CJK-space , CJK-widow } {#2}
2578   {
2579     \str_if_eq:nnTF {#3} { math }
2580     {
2581       \tl_if_eq:cnTF
2582       { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2583       {
2584         \tl_if_eq:cnTF
2585         { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2586         { \__xeCJK_boundary_use_ecglue:n {#1} }
2587         { \__xeCJK_boundary_restore_space:n {#1} }
2588       }
2589       { \__xeCJK_boundary_use_ecglue:n {#1} }
2590     }
2591     {
2592       \str_if_eq:nnTF {#3} { CJK }

```

```

2593     {
2594         \str_if_eq:nnT {#2} { CJK-widow }
2595         { \xeCJK_widow_penalty: }
2596         \tl_if_eq:cnTF
2597         { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2598         {
2599             \tl_if_eq:cnTF
2600             { g__xeCJK_boundary_capture_#1_reserve_space_flag_tl } { true }
2601             { \__xeCJK_boundary_restore_space:n {#1} }
2602             { \__xeCJK_boundary_use_ccglue:n {#1} }
2603         }
2604         { \__xeCJK_boundary_use_ccglue:n {#1} }
2605     }
2606     { \__xeCJK_boundary_use_ecglue:n {#1} }
2607 }
2608 }
2609 {
2610     \clist_if_in:nnTF { default , default-space , math } {#2}
2611     {
2612         \str_if_eq:nnTF {#3} { CJK }
2613         {
2614             \tl_if_eq:cnTF
2615             { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2616             {
2617                 \tl_if_eq:cnTF
2618                 { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2619                 { \__xeCJK_boundary_use_ecglue:n {#1} }
2620                 { \__xeCJK_boundary_restore_space:n {#1} }
2621             }
2622             { \__xeCJK_boundary_use_ecglue:n {#1} }
2623         }
2624         { \__xeCJK_boundary_restore_space:n {#1} }
2625     }
2626     { \__xeCJK_boundary_restore_space:n {#1} }
2627 }
2628 }
2629 \cs_generate_variant:Nn \__xeCJK_boundary_emit_left:nnn { nVn }
2630
2631 \cs_new_protected:Npn \__xeCJK_boundary_use_ecglue:n #1
2632 {
2633     \__xeCJK_boundary_use_glue:nn {#1}
2634     { \tl_use:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl } }
2635 }
2636
2637 \cs_new_protected:Npn \__xeCJK_boundary_use_ccglue:n #1
2638 {
2639     \__xeCJK_boundary_use_glue:nn {#1}
2640     { \tl_use:c { g__xeCJK_boundary_capture_#1_ccglue_skip_tl } }
2641 }
2642
2643 \cs_new_protected:Npn \__xeCJK_boundary_use_glue:nn #1#2
2644 {
2645     \tl_if_eq:cnTF
2646     { g__xeCJK_boundary_capture_#1_kind_tl } { stream-ulem }
2647     { \use:c { __xeCJK_boundary_use_ulem_glue:n } {#2} }
2648     { \skip_horizontal:n {#2} }
2649 }
2650
2651 \cs_new_protected:Npn \__xeCJK_boundary_restore_space:
2652 {
2653     \__xeCJK_boundary_restore_space:n
2654     { \int_use:N \g__xeCJK_boundary_capture_depth_int }
2655 }
2656
2657 \cs_new_protected:Npn \__xeCJK_boundary_restore_space:n #1
2658 {
2659     \tl_if_eq:cnT

```

```

2660     { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2661     {
2662       \__xeCJK_boundary_use_glue:nn {#1}
2663       { \tl_use:c { g__xeCJK_boundary_capture_#1_space_skip_tl } }
2664     }
2665   }
2666
2667 \cs_new_protected:Npn \__xeCJK_boundary_set_last_from_node:n #1
2668 {
2669   \clist_if_in:nnTF { CJK , CJK-space , CJK-widow } {#1}
2670   { \tl_set:Nn \l__xeCJK_boundary_last_tl { CJK } }
2671   {
2672     \clist_if_in:nnT
2673     { default , default-space } {#1}
2674     { \tl_set:Nn \l__xeCJK_boundary_last_tl { default } }
2675     \str_if_eq:nnT {#1} { math }
2676     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math } }
2677     \str_if_eq:nnT {#1} { math-space }
2678     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space } }
2679     \str_if_eq:nnT {#1} { math-space-frozen }
2680     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space-frozen } }
2681   }
2682 }
2683
2684 \cs_new_protected:Npn \__xeCJK_boundary_set_math_tail_from_capture:
2685 {
2686   \tl_if_eq:cnT
2687   {
2688     g__xeCJK_boundary_capture_
2689     \int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
2690   }
2691   { true }
2692   {
2693     \tl_if_eq:cnTF
2694     {
2695       g__xeCJK_boundary_capture_
2696       \int_use:N \g__xeCJK_boundary_capture_depth_int
2697       _math_tail_space_flag_tl
2698     }
2699     { true }
2700     {
2701       \tl_if_eq:cnTF
2702       {
2703         g__xeCJK_boundary_capture_
2704         \int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl
2705       }
2706       { stream }
2707       {
2708         \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space }
2709         \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2710         {
2711           \tl_use:c
2712           {
2713             g__xeCJK_boundary_capture_
2714             \int_use:N \g__xeCJK_boundary_capture_depth_int
2715             _math_tail_space_skip_tl
2716           }
2717         }
2718       }
2719       { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space-frozen } }
2720     }
2721     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math } }
2722   }
2723 }
2724
2725 \cs_new_protected:Npn \__xeCJK_boundary_replay_node:n #1
2726 {

```

```

2727 % Default 字符在盒子内部更新的 spacefactor 不会传播到外层列表; stream
2728 % 内容则会直接更新外层值。重放 Default 类 marker 时, 以当前列表真正的
2729 % spacefactor 同步源码空格检查上下文, 避免把盒内末尾大写字母留下的 999
2730 % 错用于盒外以 1000 生成的源码空格。
2731 \clist_if_in:nnT
2732 {
2733     default , default-space , math , math-space , math-space-frozen ,
2734     normalspace
2735 }
2736 {#1}
2737 { \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D }
2738 \str_if_eq:nnTF {#1} { math-space }
2739 {
2740     \__xeCJK_boundary_math_space_encode:
2741     \xeCJK_make_node:n {#1}
2742     \__xeCJK_boundary_enable_source_space:n {#1}
2743 }
2744 {
2745     \str_if_eq:nnTF {#1} { math-space-frozen }
2746     {
2747         \xeCJK_make_node:n {#1}
2748         \__xeCJK_boundary_enable_source_space:n {#1}
2749     }
2750     {
2751         \str_if_eq:nnTF {#1} { default-space }
2752         {
2753             \__xeCJK_make_space_node:
2754             \tl_gset:Nn \g__xeCJK_last_node_tl { default }
2755         }
2756         {
2757             \clist_if_in:nnT
2758             { CJK , CJK-space , CJK-widow , default , math , normalspace }
2759             {#1}
2760             {
2761                 \xeCJK_make_node:n {#1}
2762                 \__xeCJK_boundary_enable_source_space:n {#1}
2763             }
2764         }
2765     }
2766 }
2767 }
2768 \cs_new_protected:Npn \__xeCJK_boundary_enable_source_space:n #1
2769 {
2770     \clist_if_in:nnT
2771     {
2772         CJK , CJK-space , CJK-widow , math , math-space ,
2773         math-space-frozen
2774     }
2775     {#1}
2776     { \bool_gset_true:N \g__xeCJK_glue_check_pending_bool }
2777 }
2778
2779 \cs_new_protected:Npn \__xeCJK_boundary_register_box:nn #1#2
2780 {
2781     \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2782     {
2783         \prop_gput:Nnn \g__xeCJK_boundary_registered_prop {#1} {#2}
2784         \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2785         { \__xeCJK_boundary_inline_last_box_begin: }
2786         \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2787         { \__xeCJK_boundary_inline_last_box_end:n {#2} }
2788     }
2789 }
2790
2791 \cs_new_protected:Npn \__xeCJK_boundary_register_wrapped_box:nn #1#2
2792 {
2793     \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}

```

```

2794     {
2795         \prop_gput:Nnn
2796         \g__xeCJK_boundary_registered_prop {#1} { wrapped-box/#2 }
2797         \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2798         { \__xeCJK_boundary_inline_box_begin: }
2799         \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2800         { \__xeCJK_boundary_inline_box_end:n {#2} }
2801     }
2802 }
2803
2804 \cs_new_protected:Npn \__xeCJK_boundary_register_transparent:n #1
2805 {
2806     \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2807     {
2808         \prop_gput:Nnn
2809         \g__xeCJK_boundary_registered_prop {#1} { transparent }
2810         \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2811         { \__xeCJK_boundary_hmode_transparent_begin: }
2812         \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2813         { \__xeCJK_boundary_hmode_transparent_end: }
2814     }
2815 }
2816
2817 \cs_new_protected:Npn \__xeCJK_boundary_register_post_transparent:n #1
2818 {
2819     \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2820     {
2821         \prop_gput:Nnn
2822         \g__xeCJK_boundary_registered_prop {#1} { post-transparent }
2823         \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2824         { \__xeCJK_boundary_post_transparent: }
2825     }
2826 }
2827
2828 \cs_new_protected:Npn \__xeCJK_boundary_register_stream:nn #1#2
2829 {
2830     \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2831     {
2832         \prop_gput:Nnn \g__xeCJK_boundary_registered_prop {#1} { stream/#2 }
2833         \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2834         { \__xeCJK_boundary_inline_stream_begin:n {#2} }
2835         \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2836         { \__xeCJK_boundary_inline_stream_end:n {#2} }
2837     }
2838 }
2839
2840 % 通用注册函数把命令写入 \cs{g__xeCJK_boundary_registered_prop}。另有一些
2841 % 命令必须直接重定义内部排版入口，不能使用 LaTeX 通用命令 hook；这些
2842 % 入口写入独立的保留表。用户接口同时查询两张表，防止把通用 hook 叠加到
2843 % 已有专用适配器外层。
2844 \cs_new_protected:Npn \__xeCJK_boundary_reserve:n #1
2845 { \prop_gput:Nnn \g__xeCJK_boundary_reserved_prop {#1} { adapter } }
2846 \cs_new_protected:Npn \__xeCJK_boundary_reserve:N #1
2847 {
2848     \exp_args:Ne \__xeCJK_boundary_reserve:n { \cs_to_str:N #1 }
2849 }
2850 \prg_new_conditional:Npnn \__xeCJK_boundary_if_registered:n #1 { T , F , TF }
2851 {
2852     \prop_if_in:NnTF \g__xeCJK_boundary_registered_prop {#1}
2853     { \prg_return_true: }
2854     {
2855         \prop_if_in:NnTF \g__xeCJK_boundary_reserved_prop {#1}
2856         { \prg_return_true: }
2857         { \prg_return_false: }
2858     }
2859 }
2860

```

```

2861 \__xeCJK_msg_new:nn { boundary-register-command }
2862 {
2863   The~value~of~`command'~in~`experiment/boundary-register'~must~be~
2864   exactly~one~control~sequence.
2865 }
2866 \__xeCJK_msg_new:nn { boundary-register-choice }
2867 { Invalid~boundary~registration~choice:~#1. }
2868 \__xeCJK_msg_new:nn { boundary-register-missing }
2869 { The~`#1'~entry~is~required~in~`experiment/boundary-register'. }
2870 \__xeCJK_msg_new:nn { boundary-register-mode }
2871 { The~boundary~registration~strategy~and~mode~cannot~be~combined:~#1. }
2872 \__xeCJK_msg_new:nn { boundary-register-duplicate }
2873 { The~command~`#1'~has~already~been~declared~for~boundary~registration. }
2874 \__xeCJK_msg_new:nn { boundary-register-conflict }
2875 { The~command~`#1'~already~has~an~xeCJK~boundary~registration. }
2876 \__xeCJK_msg_new:nn { boundary-register-late }
2877 {
2878   The~`experiment/boundary-register'~option~can~only~be~used~before~
2879   the~end~of~the~preamble.
2880 }
2881 \__xeCJK_msg_new:nn { boundary-register-undefined }
2882 { The~registered~boundary~command~`#1'~is~still~undefined. }
2883
2884 \cs_new_protected:Npn \__xeCJK_boundary_user_register:n #1
2885 {
2886   \bool_if:NTF \g__xeCJK_boundary_user_register_closed_bool
2887   { \__xeCJK_error:n { boundary-register-late } }
2888   {
2889     \bool_set_false:N \l__xeCJK_boundary_user_invalid_bool
2890     \bool_set_false:N \l__xeCJK_boundary_user_mode_given_bool
2891     \tl_clear:N \l__xeCJK_boundary_user_name_tl
2892     \tl_clear:N \l__xeCJK_boundary_user_strategy_tl
2893     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { auto }
2894     \keys_set:nn { xeCJK / boundary-register } { #1 }
2895     \tl_if_empty:NT \l__xeCJK_boundary_user_name_tl
2896     {
2897       \__xeCJK_error:ne { boundary-register-missing } { command }
2898       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2899     }
2900     \tl_if_empty:NT \l__xeCJK_boundary_user_strategy_tl
2901     {
2902       \__xeCJK_error:ne { boundary-register-missing } { strategy }
2903       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2904     }
2905     \bool_if:NF \l__xeCJK_boundary_user_invalid_bool
2906     { \__xeCJK_boundary_user_register_validate: }
2907     \bool_if:NF \l__xeCJK_boundary_user_invalid_bool
2908     { \__xeCJK_boundary_user_register_store: }
2909   }
2910 }
2911
2912 \cs_new_protected:Npn \__xeCJK_boundary_user_register_command:n #1
2913 {
2914   \tl_if_single_token:nTF { #1 }
2915   {
2916     \token_if_cs:NTF #1
2917     { \tl_set:Ne \l__xeCJK_boundary_user_name_tl { \cs_to_str:N #1 } }
2918     {
2919       \__xeCJK_error:n { boundary-register-command }
2920       \tl_set:Nn \l__xeCJK_boundary_user_name_tl { invalid }
2921       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2922     }
2923   }
2924   {
2925     \__xeCJK_error:n { boundary-register-command }
2926     \tl_set:Nn \l__xeCJK_boundary_user_name_tl { invalid }
2927     \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool

```

```

2928     }
2929 }
2930
2931 \cs_new_protected:Npn \__xeCJK_boundary_user_register_strategy:n #1
2932 {
2933     \str_case:nnF {#1}
2934     {
2935         { box }                { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2936         { wrapped-box }        { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2937         { stream }              { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2938         { transparent }         { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2939         { post-transparent }    { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2940     }
2941     {
2942         \__xeCJK_error:ne { boundary-register-choice }
2943         { strategy~=~\tl_to_str:n {#1} }
2944         \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl { invalid }
2945         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2946     }
2947 }
2948
2949 \cs_new_protected:Npn \__xeCJK_boundary_user_register_mode:n #1
2950 {
2951     \bool_set_true:N \l__xeCJK_boundary_user_mode_given_bool
2952     \str_case:nnF {#1}
2953     {
2954         { auto }                { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
2955         { default }              { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
2956         { first-default }        { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
2957     }
2958     {
2959         \__xeCJK_error:ne { boundary-register-choice }
2960         { mode~=~\tl_to_str:n {#1} }
2961         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2962     }
2963 }
2964
2965 \cs_new_protected:Npn \__xeCJK_boundary_user_register_validate:
2966 {
2967     \str_case:Vn \l__xeCJK_boundary_user_strategy_tl
2968     {
2969         { box }
2970         {
2971             \tl_if_eq:NnT \l__xeCJK_boundary_user_mode_tl { first-default }
2972             {
2973                 \__xeCJK_error:ne { boundary-register-mode } { box/first-default }
2974                 \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2975             }
2976         }
2977         { wrapped-box }
2978         {
2979             \tl_if_eq:NnT \l__xeCJK_boundary_user_mode_tl { first-default }
2980             {
2981                 \__xeCJK_error:ne { boundary-register-mode }
2982                 { wrapped-box/first-default }
2983                 \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2984             }
2985         }
2986         { stream } { }
2987         { transparent }
2988         {
2989             \bool_if:NT \l__xeCJK_boundary_user_mode_given_bool
2990             {
2991                 \__xeCJK_error:ne { boundary-register-mode }
2992                 { transparent/\l__xeCJK_boundary_user_mode_tl }
2993                 \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2994             }

```

```

2995         \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { none }
2996     }
2997     { post-transparent }
2998     {
2999         \bool_if:NT \l__xeCJK_boundary_user_mode_given_bool
3000         {
3001             \__xeCJK_error:ne { boundary-register-mode }
3002             { post-transparent/\l__xeCJK_boundary_user_mode_tl }
3003             \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3004         }
3005         \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { none }
3006     }
3007 }
3008 }
3009
3010 \cs_new_protected:Npn \__xeCJK_boundary_user_register_store:
3011 {
3012     \exp_args:NV \__xeCJK_boundary_if_registered:nTF
3013     \l__xeCJK_boundary_user_name_tl
3014     {
3015         \__xeCJK_error:ne { boundary-register-conflict }
3016         { \c_backslash_str \l__xeCJK_boundary_user_name_tl }
3017     }
3018     {
3019         \exp_args:NNV \prop_if_in:NnTF
3020         \g__xeCJK_boundary_user_registered_prop \l__xeCJK_boundary_user_name_tl
3021         {
3022             \__xeCJK_error:ne { boundary-register-duplicate }
3023             { \c_backslash_str \l__xeCJK_boundary_user_name_tl }
3024         }
3025         {
3026             \tl_set:Nn \l__xeCJK_boundary_user_value_tl
3027             {
3028                 \l__xeCJK_boundary_user_strategy_tl
3029                 / \l__xeCJK_boundary_user_mode_tl
3030             }
3031             \exp_args:NNVV \prop_gput:Nnn
3032             \g__xeCJK_boundary_user_registered_prop
3033             \l__xeCJK_boundary_user_name_tl
3034             \l__xeCJK_boundary_user_value_tl
3035         }
3036     }
3037 }
3038
3039 \cs_new_protected:Npn \__xeCJK_boundary_user_register_apply:
3040 {
3041     \bool_gset_true:N \g__xeCJK_boundary_user_register_closed_bool
3042     \prop_map_inline:Nn \g__xeCJK_boundary_user_registered_prop
3043     {
3044         \__xeCJK_boundary_if_registered:nTF {##1}
3045         {
3046             \__xeCJK_error:ne { boundary-register-conflict }
3047             { \c_backslash_str ##1 }
3048         }
3049         {
3050             \__xeCJK_boundary_user_register_apply:nn {##1} {##2}
3051             \seq_gput_right:Nn \g__xeCJK_boundary_user_applied_seq {##1}
3052         }
3053     }
3054 }
3055
3056 \cs_new_protected:Npn \__xeCJK_boundary_user_register_apply:nn #1#2
3057 {
3058     \seq_set_split:Nnn \l__xeCJK_boundary_user_value_seq { / } {#2}
3059     \seq_pop_left:NN
3060     \l__xeCJK_boundary_user_value_seq \l__xeCJK_boundary_user_strategy_tl
3061     \seq_pop_left:NN

```

```

3062 \l__xeCJK_boundary_user_value_seq \l__xeCJK_boundary_user_mode_tl
3063 \str_case:Vn \l__xeCJK_boundary_user_strategy_tl
3064 {
3065   { box }
3066   {
3067     \exp_args:NnV \__xeCJK_boundary_register_box:nn
3068     {#1} \l__xeCJK_boundary_user_mode_tl
3069   }
3070   { wrapped-box }
3071   {
3072     \exp_args:NnV \__xeCJK_boundary_register_wrapped_box:nn
3073     {#1} \l__xeCJK_boundary_user_mode_tl
3074   }
3075   { stream }
3076   {
3077     \exp_args:NnV \__xeCJK_boundary_register_stream:nn
3078     {#1} \l__xeCJK_boundary_user_mode_tl
3079   }
3080   { transparent }
3081   { \__xeCJK_boundary_register_transparent:n {#1} }
3082   { post-transparent }
3083   { \__xeCJK_boundary_register_post_transparent:n {#1} }
3084 }
3085 }
3086
3087 \cs_new_protected:Npn \__xeCJK_boundary_user_register_check:
3088 {
3089   \seq_map_inline:Nn \g__xeCJK_boundary_user_applied_seq
3090   {
3091     \cs_if_exist:cF {##1}
3092     {
3093       \__xeCJK_error:ne { boundary-register-undefined }
3094       { \c_backslash_str ##1 }
3095     }
3096   }
3097 }
3098
3099 % \tn{mbox} 和 \tn{fbox} 的公开命令是 robust wrapper; 实际单参数正文由带
3100 % 尾随空格的内部控制序列接收。必须先包装这层, 再安装公开命令的 before
3101 % hook: 这样 hook 先启动 box capture, 正文适配器随后才能报告开头和结尾
3102 % 的公式类别。适配器只检查用户交给盒子命令的可见正文, 不监听命令内部的
3103 % \tn{setbox} 测量。
3104 \cs_new_protected:Npn \__xeCJK_boundary_prepare_onearg_box:N #1
3105 {
3106   \group_begin: \exp_args:NcNc \group_end:
3107   { \__xeCJK_boundary_prepare_onearg_box:NN }
3108   #1 { \cs_to_str:N #1 ~ }
3109 }
3110 \cs_new_protected:Npn \__xeCJK_boundary_prepare_onearg_box:NN #1#2
3111 {
3112   \__xeCJK_boundary_reserve:N #2
3113   \exp_args:Nc \cs_new_eq:NN
3114   { \__xeCJK_boundary_orig_\cs_to_str:N #1 _box:n } #2
3115   \cs_gset_protected:Npn #2 ##1
3116   {
3117     \__xeCJK_boundary_math_begin:n {##1}
3118     \use:c { \__xeCJK_boundary_orig_\cs_to_str:N #1 _box:n }
3119     { ##1 \__xeCJK_boundary_math_end:n {##1} }
3120   }
3121 }
3122
3123 % expl3 内部函数不一定允许 LaTeX 通用 cmd hook。下面两个适配器仍复用
3124 % transparent capture 的同一 begin/end, 只负责保留零参数或单参数调用签名。
3125 \cs_new_protected:Npn \__xeCJK_boundary_wrap_transparent_noarg:NN #1#2
3126 {
3127   \cs_if_exist:NF #1
3128   {

```

```

3129     \__xeCJK_boundary_reserve:N #2
3130     \cs_set_eq:NN #1#2
3131     \cs_gset_protected:Npn #2
3132     {
3133         \__xeCJK_boundary_hmode_transparent_begin:
3134         #1
3135         \__xeCJK_boundary_hmode_transparent_end:
3136     }
3137 }
3138 }
3139
3140 \cs_new_protected:Npn \__xeCJK_boundary_wrap_transparent_onearg:NN #1#2
3141 {
3142     \cs_if_exist:NF #1
3143     {
3144         \__xeCJK_boundary_reserve:N #2
3145         \cs_set_eq:NN #1#2
3146         \cs_gset_protected:Npn #2 ##1
3147         {
3148             \__xeCJK_boundary_hmode_transparent_begin:
3149             #1 ##1
3150             \__xeCJK_boundary_hmode_transparent_end:
3151         }
3152     }
3153 }
3154
3155 % \changes{v3.10.4}{2026/07/20}{包装内核私有 \tn{@imakebox} 与
3156 % \tn{@iframebox} 的 |[\#1][\#2]\#3| 签名以注册 \tn{makebox} /
3157 % \tn{framebox}; 内核若调整该签名须同步此 wrapper (\#992)。}
3158 % 这里包装的是本包最低支持内核 \LaTeXe{} 2026-06-01 中, 内部
3159 % \tn{@imakebox}/\tn{@iframebox} 的 |[\#1][\#2]\#3| 参数签名, 不是稳定公共
3160 % 接口。若内核改变该签名, 必须同步 wrapper; \texttt{command-boundary01}
3161 % 的 optional \tn{makebox}/\tn{framebox} 场景作为漂移门禁。
3162 \cs_new_protected:Npn \__xeCJK_boundary_register_makeboxes:
3163 {
3164     \clist_map_inline:nn { makebox , framebox , @imakebox , @iframebox }
3165     { \__xeCJK_boundary_reserve:n {##1} }
3166     \cs_if_exist:NF \__xeCJK_orig_imakebox:w
3167     {
3168         \cs_new_eq:NN \__xeCJK_orig_imakebox:w \@imakebox
3169         \cs_gset_protected:Npn \@imakebox [##1][##2]##3
3170         {
3171             \__xeCJK_boundary_inline_last_box_begin:
3172             \__xeCJK_boundary_math_begin:n {##3}
3173             \__xeCJK_orig_imakebox:w [##1][##2]
3174             { ##3 \__xeCJK_boundary_math_end:n {##3} }
3175             \__xeCJK_boundary_inline_last_box_end:n { auto }
3176         }
3177     }
3178     \cs_if_exist:NF \__xeCJK_orig_iframebox:w
3179     {
3180         \cs_new_eq:NN \__xeCJK_orig_iframebox:w \@iframebox
3181         \cs_gset_protected:Npn \@iframebox [##1][##2]##3
3182         {
3183             \__xeCJK_boundary_inline_last_box_begin:
3184             \__xeCJK_boundary_math_begin:n {##3}
3185             \__xeCJK_orig_iframebox:w [##1][##2]
3186             { ##3 \__xeCJK_boundary_math_end:n {##3} }
3187             \__xeCJK_boundary_inline_last_box_end:n { auto }
3188         }
3189     }
3190 }
3191
3192 \__xeCJK_boundary_prepare_onearg_box:N \mbox
3193 \__xeCJK_boundary_prepare_onearg_box:N \fbox
3194 \__xeCJK_boundary_register_box:nn { mbox } { auto }
3195 \__xeCJK_boundary_register_box:nn { fbox } { auto }

```

```

3196 \__xeCJK_boundary_register_makeboxes:
3197 \__xeCJK_boundary_register_post_transparent:n { null }
3198 \__xeCJK_boundary_reserve:n { sbox }
3199 \AddToHook { cmd / sbox / before } [ xeCJK / boundary ]
3200 { \__xeCJK_boundary_capture_suspend: }
3201 \AddToHook { cmd / sbox / after } [ xeCJK / boundary ]
3202 { \__xeCJK_boundary_capture_resume: }

```

\xeCJK_make_space_node: 用于判断插入空格之前的 node, 默认为空, 只有用户设置了 xCJKecglue 选项才有意义。需要使用 glue 来标记, 使用 kern 会影响 character protrusion 功能。

```

3203 \cs_new_eq:NN \xeCJK_make_space_node: \prg_do_nothing:
3204 \cs_new_protected:Npe \__xeCJK_make_space_node:
3205 {
3206   \tex_hskip:D - \__xeCJK_glue_node:n { default-space }
3207   \tex_hskip:D \__xeCJK_glue_node:n { default-space }
3208 }

```

CJKglue CJK 文字之间插入的 glue。

```

3209 \keys_define:nn { xeCJK / options }
3210 {
3211   CJKglue .code:n =
3212   {
3213     \cs_set_protected:Npn \CJKglue {#1}
3214     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
3215   }
3216 }
3217 \skip_new:N \l__xeCJK_ccglue_skip

```

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

```

3218 \keys_define:nn { xeCJK / options }
3219 {
3220   CJKecglue .code:n =
3221   {
3222     \cs_set_protected:Npn \CJKecglue {#1}
3223     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
3224   } ,
3225   xCJKecglue .choice: ,
3226   xCJKecglue / true .code:n =
3227   {
3228     \bool_set_true:N \l__xeCJK_xecglue_bool
3229     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
3230     \cs_set_eq:NN \xeCJK_make_space_node: \__xeCJK_make_space_node:
3231     \cs_set_eq:NN \xeCJK_check_for_xglue: \__xeCJK_check_for_xglue:
3232     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_xecglue:
3233     \cs_set_eq:NN
3234     \xeCJK_check_for_ecglue_normalsp:
3235     \__xeCJK_check_for_xecglue_normalsp:
3236   } ,
3237   xCJKecglue / false .code:n =
3238   {
3239     \bool_set_false:N \l__xeCJK_xecglue_bool
3240     \cs_set_eq:NN \xeCJK_space_or_xecglue: \xeCJK_space_glue:
3241     \xeCJK_cs_clear:N \xeCJK_make_space_node:
3242     \xeCJK_cs_clear:N \xeCJK_check_for_xglue:
3243     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:
3244     \cs_set_eq:NN
3245     \xeCJK_check_for_ecglue_normalsp:
3246     \__xeCJK_check_for_ecglue_normalsp:
3247   } ,
3248   xCJKecglue / unknown .code:n =
3249   {
3250     \bool_set_true:N \l__xeCJK_xecglue_bool
3251     \cs_set_protected:Npn \CJKecglue {#1}

```

```

3252 \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
3253 \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKeCglue
3254 \cs_set_eq:NN \xeCJK_make_space_node: \l__xeCJK_make_space_node:
3255 \cs_set_eq:NN \xeCJK_check_for_xglue: \l__xeCJK_check_for_xglue:
3256 \cs_set_eq:NN \xeCJK_check_for_ecglue: \l__xeCJK_check_for_xecglue:
3257 \cs_set_eq:NN
3258 \xeCJK_check_for_ecglue_normalsp:
3259 \l__xeCJK_check_for_xecglue_normalsp:
3260 } ,
3261 xCJKeCglue .default:n = { true }
3262 }
3263 \cs_new_eq:NN \xeCJK_space_glue: \c_space_tl
3264 \skip_new:N \l__xeCJK_ecglue_skip
3265 \bool_new:N \l__xeCJK_xecglue_bool

```

CJKspace 是否保留 CJK 文字间的空白, 默认不保留。

```

3266 \keys_define:nn { xeCJK / options }
3267 {
3268   CJKspace .choice: ,
3269   CJKspace / true .code:n =
3270   {
3271     \bool_set_true:N \l__xeCJK_reserve_space_bool
3272     \cs_set_protected:Npn \l__xeCJK_ccglue_or_space:
3273     { \xeCJK_space_glue: }
3274   } ,
3275   CJKspace / false .code:n =
3276   {
3277     \bool_set_false:N \l__xeCJK_reserve_space_bool
3278     \cs_set_protected:Npn \l__xeCJK_ccglue_or_space:
3279     { \CJKeCglue }
3280   } ,
3281   CJKspace .default:n = { true } ,
3282   space .meta:n = { CJKspace = true } ,
3283   nospace .meta:n = { CJKspace = false }
3284 }
3285 \bool_new:N \l__xeCJK_reserve_space_bool
3286 \xeCJK_inter_class_toks:nnn { CJK } { Boundary } { \xeCJK_CJK_and_Boundary:w }

```

\xeCJK_CJK_and_Boundary:w 当边界是 \relax 的时候, 它可能是由 \csname ... \endcsname 的形式产生的, 这样就可能出现¹⁵。原来是都在未定义控制序列前都加上 \exp_not:N, 现在则采用分组结束后手工恢复的方式。

```

3287 \cs_new_protected:Npn \xeCJK_CJK_and_Boundary:w
3288 {
3289   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
3290   { \l__xeCJK_CJK_and_math: }
3291   {
3292     \group_align_safe_begin:
3293     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3294     {
3295       \token_if_macro:NTF \l_peek_token
3296       {
3297         \token_if_eq_meaning:NNTF
3298         \l_peek_token \l__xeCJK_boundary_identity:n
3299         { \l__xeCJK_boundary_peek_math:w }
3300         { \l__xeCJK_boundary_reserve_space: }
3301       }
3302       {
3303         \token_if_group_begin:NTF \l_peek_token
3304         { \l__xeCJK_boundary_group_math:w }
3305         { \l__xeCJK_boundary_group_end:n { CJK-space } }
3306       }
3307     }

```

¹⁵ 参见 <http://bbs.ctex.org/forum.php?mod=viewthread&tid=71563>。

```

3308     {
3309         \token_if_eq_meaning:NNTF \l_peek_token \scan_stop:
3310         { \__xeCJK_CJK_and_Boundary_relax:N }
3311         {
3312             \token_if_group_begin:NNTF \l_peek_token
3313             { \__xeCJK_boundary_group_math:w }
3314             {
3315                 \token_if_group_end:NNTF \l_peek_token
3316                 {
3317                     \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
3318                     \__xeCJK_boundary_group_end:n { CJK }
3319                 }
3320                 {
3321                     \token_if_macro:NNTF \l_peek_token
3322                     {
3323                         \token_if_eq_meaning:NNTF
3324                         \l_peek_token \__xeCJK_boundary_identity:n
3325                         { \__xeCJK_boundary_peek_math:w }
3326                         { \__xeCJK_boundary_group_end:n { CJK } }
3327                     }
3328                     { \__xeCJK_boundary_group_end:n { CJK } }
3329                 }
3330             }
3331         }
3332     }
3333 }
3334 }
3335 \cs_new_protected:Npn \__xeCJK_CJK_and_math:
3336 {
3337     \bool_if:NNTF \l__xeCJK_peek_ignore_spaces_bool
3338     { \xeCJK_class_group_end: \xeCJK_space_or_xecglue: }
3339     { \xeCJK_class_group_end: \CJKecglue }
3340 }
3341 \cs_new_protected:Npn \__xeCJK_boundary_CJK_and_math:
3342 { \group_align_safe_end: \__xeCJK_CJK_and_math: }
3343 \cs_new_protected:Npn \__xeCJK_boundary_group_math:w #1
3344 {
3345     \tl_if_empty:nTF {#1}
3346     {
3347         \bool_if:NNTF \l__xeCJK_peek_ignore_spaces_bool
3348         { \__xeCJK_boundary_reserve_space: }
3349         { \__xeCJK_boundary_group_end:n { CJK } }
3350     }
3351     {
3352         \tl_if_head_eq_meaning:nNTF {#1} $
3353         { \__xeCJK_boundary_CJK_and_math: }
3354         {
3355             \bool_if:NNTF \l__xeCJK_peek_ignore_spaces_bool
3356             { \__xeCJK_boundary_reserve_space: }
3357             { \__xeCJK_boundary_group_end:n { CJK } }
3358         }
3359     }
3360     \c_group_begin_token #1 \c_group_end_token
3361 }
3362 % 只展开严格等同于单参数原样输出函数的宏。对任意“可展开”控制序列做
3363 % 预展开会提前执行条件判断或扫描参数，破坏表格等正在运行的 TeX 语法。
3364 \cs_new:Npn \__xeCJK_boundary_identity:n #1 {#1}
3365 \cs_new_protected:Npn \__xeCJK_boundary_peek_math:w
3366 {
3367     \exp_after:wN \peek_after:Nw
3368     \exp_after:wN \__xeCJK_boundary_peek_math_branches:w
3369     \exp:w \exp_end_continue_f:w
3370 }
3371 \cs_new_protected:Npn \__xeCJK_boundary_peek_math_branches:w
3372 {
3373     \token_if_math_toggle:NNTF \l_peek_token
3374     { \__xeCJK_boundary_CJK_and_math: }

```

```

3375     {
3376       \token_if_group_begin:NTF \l_peek_token
3377       { \__xeCJK_boundary_group_math:w }
3378       {
3379         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3380         { \__xeCJK_boundary_reserve_space: }
3381         { \__xeCJK_boundary_group_end:n { CJK } }
3382       }
3383     }
3384   }
3385   \cs_new_protected:Npn \__xeCJK_boundary_reserve_space:
3386   { \__xeCJK_boundary_group_end:n { CJK-space } }
3387   \cs_new_protected:Npn \__xeCJK_CJK_and_Boundary_relax:N #1
3388   {
3389     \__xeCJK_boundary_group_end:n { CJK }
3390     \token_if_eq_meaning:NNTF #1 \scan_stop:
3391     { #1 } { \cs_set_eq:NN #1 \scan_stop: #1 }
3392   }
3393   \cs_new_protected:Npn \__xeCJK_boundary_group_end:n #1
3394   {
3395     \group_align_safe_end:
3396     \xeCJK_class_group_end:
3397     \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
3398     { \xeCJK_make_node:n { #1 } }
3399   }

\xeCJK_ignore_spaces:w 3400 \cs_new_protected:Npn \xeCJK_ignore_spaces:w
3401   {
3402     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
3403     {
3404       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3405       { \xeCJK_space_or_xecglue: } { \CJKecglue }
3406     }
3407     {
3408       \bool_if:NT \l__xeCJK_peek_ignore_spaces_bool
3409       {
3410         \dim_case:nn { \tex_lastkern:D }
3411         {
3412           { \__xeCJK_node:n { CJK } }
3413           { \xeCJK_remove_node: \xeCJK_make_node:n { CJK-space } }
3414           { \__xeCJK_node:n { default } }
3415           { \xeCJK_remove_node: \xeCJK_make_space_node: }
3416         }
3417         \group_align_safe_begin:
3418         \token_if_macro:NTF \l_peek_token
3419         { \__xeCJK_reserve_space_aux: }
3420         { \group_align_safe_end: }
3421       }
3422     }
3423   }
3424   \cs_new_protected:Npn \__xeCJK_reserve_space_aux:
3425   {
3426     \group_align_safe_end:
3427     \xeCJK_space_or_xecglue:
3428   }

3429   \xeCJK_inter_class_toks:nnn { CJK } { CJK }
3430   { \xeCJK_CJK_and_CJK:N }

\xeCJK_CJK_and_CJK:N 3431 \cs_new_protected:Npn \xeCJK_CJK_and_CJK:N
3432   {
3433     \CJKglue
3434     \xeCJK_fallback_symbol:NN
3435     \CJKsymbol
3436   }

3437   \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
3438   {

```

```

3439 \xeCJK_FullLeft_and_CJK:
3440 \xeCJK_fallback_symbol:NN
3441 \CJKsymbol
3442 }
3443 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
3444 {
3445 \xeCJK_FullRight_and_CJK:
3446 \xeCJK_fallback_symbol:NN
3447 \CJKsymbol
3448 }
3449 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
3450 {
3451 \clist_map_inline:nn { FullLeft , FullRight }
3452 {
3453 \xeCJK_inter_class_toks:nne {#1} {##1}
3454 { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
3455 \xeCJK_inter_class_toks:nne {##1} {#1}
3456 { \exp_not:c { xeCJK_##1_and_Default: } }
3457 }
3458 }

```

```
\l__xeCJK_halfright_nobreak_bool 3459 \bool_new:N \l__xeCJK_halfright_nobreak_bool
```

在 CJK → HalfRight 和 FullRight → HalfRight 的过渡中,条件性地插入禁则 penalty。

```

3460 \xeCJK_app_inter_class_toks:nnn { CJK } { HalfRight }
3461 {
3462 \bool_if:NT \l__xeCJK_halfright_nobreak_bool
3463 { \xeCJK_no_break: }
3464 }

```

FullRight → HalfRight 需要将 penalty 插入 punct_glue 之前,因此覆写而非追加。

```

3465 \xeCJK_inter_class_toks:nnn { FullRight } { HalfRight }
3466 {
3467 \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3468 \xeCJK_class_group_end:
3469 \bool_if:NT \l__xeCJK_halfright_nobreak_bool
3470 { \xeCJK_no_break: }
3471 \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3472 }
3473 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
3474 { \xeCJK_Boundary_and_FullLeft:N }
3475 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
3476 { \xeCJK_Boundary_and_FullRight:N }
3477 \xeCJK_inter_class_toks:nnn { FullLeft } { Boundary }
3478 { \xeCJK_FullLeft_and_Boundary: }
3479 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
3480 { \xeCJK_FullRight_and_Boundary: }

```

```

\l__xeCJK_FullLeft_and_Boundary: 3481 \cs_new_protected:Npn \xeCJK_FullLeft_and_Boundary:
3482 {
3483 \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3484 {
3485 \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3486 \xeCJK_class_group_end:
3487 \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3488 \xeCJK_no_break:
3489 \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3490 \__xeCJK_punct_boundary_guard:
3491 }
3492 {
3493 \xeCJK_class_group_end:
3494 \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3495 \__xeCJK_nobreak_zero_glue:
3496 }
3497 \tex_ignorespaces:D
3498 }

```

```

\xCJK_FullRight_and_Boundary: 3499 \cs_new_protected:Npn \xeCJK_FullRight_and_Boundary:
3500 {
3501   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3502   \xeCJK_class_group_end:
3503   \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3504   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3505   \__xeCJK_punct_boundary_guard:
3506   \tex_ignorespaces:D
3507 }

```

`\__xeCJK_punct_boundary_guard:` 在 restricted horizontal mode (tabular 单元格、\hbox 等) 中, \结束行时通过 \unskip 移除最后一个 glue。这会吞掉标点的补偿 glue, 导致单元格宽度错误。此函数在 inner mode 下于 glue 之后插入 \penalty 0, 使最后节点不再是 glue, 从而保护它。

在段落模式下, 当 experiment/punct-measure-fix 选项启用时, 记录 glue 的自然宽度, 并通过 para/end 钩子插入等宽 \kern 恢复。使用 \kern 而非 \hskip: \kern 不会被 \unskip 移除 (无需额外保护), 也不构成合法断行点 (避免 CheckSingle 等场景下产生空行)。

```

3508 \bool_new:N \g__xeCJK_par_guard_bool
3509 \dim_new:N \g__xeCJK_par_guard_dim
3510 \cs_new_protected:Npn \__xeCJK_punct_boundary_guard:
3511 {
3512   \mode_if_inner:TF
3513   { \tex_penalty:D \c_zero_int }
3514   {
3515     \bool_if:NT \g__xeCJK_punct_measure_fix_bool
3516     {
3517       \bool_gset_true:N \g__xeCJK_par_guard_bool
3518       \dim_gset:Nn \g__xeCJK_par_guard_dim { \tex_lastskip:D }
3519     }
3520   }
3521 }

```

当 Boundary 之后紧跟其他字符类时, 说明标点不在段末, 需要重置保护标志。

```

3522 \clist_map_inline:nn
3523 {
3524   Default , HalfLeft , HalfRight , NormalSpace ,
3525   CJK , FullLeft , FullRight , CM , HangulJamo ,
3526   HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
3527 }
3528 {
3529   \xeCJK_pre_inter_class_toks:nnn { Boundary } {#1}
3530   { \bool_gset_false:N \g__xeCJK_par_guard_bool }
3531 }

```

在 para/begin 中重置标志, 避免跨段落残留状态。在 para/end 中, 若标志为真则插入 \kern 补偿被 \unskip 移除的标点 glue 的自然宽度。

```

3532 \AddToHook { para/begin }
3533 { \bool_gset_false:N \g__xeCJK_par_guard_bool }
3534 \AddToHook { para/end }
3535 {
3536   \bool_if:NT \g__xeCJK_par_guard_bool
3537   {
3538     \tex_kern:D \g__xeCJK_par_guard_dim
3539     \bool_gset_false:N \g__xeCJK_par_guard_bool
3540   }
3541 }

```

`\xeCJK_punct_node:N` 保存标点的当前边界宽度和字符码, 通过插入 \kern 实现。

```

3542 \cs_new_protected:Npn \xeCJK_punct_node:N #1
3543 {
3544   \__xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
3545   \__xeCJK_make_node:N \l__xeCJK_tmp_dim
3546   \dim_set:Nn \l__xeCJK_tmp_dim { `#1 sp }
3547   \__xeCJK_make_node:N \l__xeCJK_tmp_dim
3548 }

```

_xeCJK_punct_bound_unitization:NN 我们不想出现过大的 \kern, 因此当边界大于 1pt 时, 以 \c_max_dim 为标准对其进行“单位化”。

```

3549 \cs_new_protected:Npn \_xeCJK_punct_bound_unitization:NN #1#2
3550 {
3551   \dim_set:Nn #2
3552   {
3553     \dim_max:nn
3554     { \c_zero_dim }
3555     { \_xeCJK_use_punct_dim:nn { bound } \c_xeCJK_right_tl #1 }
3556   }
3557   \dim_compare:nNf {#2} < { 1pt }
3558   { \dim_set:Nn #2 { -1pt * \dim_ratio:nn {#2} { \c_max_dim } } }
3559 }

```

```

\_xeCJK_punct_bound_kern:N 3560 \cs_new_protected:Npn \xeCJK_punct_bound_kern:N #1
\_xeCJK_punct_bound_kern:NN 3561 {
3562   \exp_after:wN \_xeCJK_punct_bound_kern:NN
3563   \g_xeCJK_last_punct_tl #1
3564 }
3565 \cs_new_protected:Npn \_xeCJK_punct_bound_kern:NN #1#2
3566 {
3567   \xeCJK_get_punct_bounds:NN \l__xeCJK_aligni_tl #1
3568   \xeCJK_get_punct_kerning:NN #1 #2
3569   \_xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
3570   \skip_set:Nn \l__xeCJK_punct_kern_skip
3571   { \_xeCJK_use_dim_or_skip:nn { bound_kern } #1 #2 }
3572   \dim_compare:nNf \l__xeCJK_tmp_dim = \l__xeCJK_last_bound_dim
3573   { \_xeCJK_punct_bound_kern_ratio:NN #1 #2 }
3574   \bool_if:NTF \l__xeCJK_last_penalty_bool
3575   {
3576     \tex_penalty:D \l__xeCJK_last_penalty_int
3577     \skip_horizontal:N
3578   }
3579   { \_xeCJK_punct_bound_kern_aux:NN #1 #2 }
3580   \l__xeCJK_punct_kern_skip
3581 }
3582 \skip_new:N \l__xeCJK_punct_kern_skip

```

_xeCJK_punct_bound_kern_ratio:NN 当标点前后的字体情况不一致时, 按一定的比例进行压缩。

```

3583 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_ratio:NN #1#2
3584 {
3585   \dim_set:Nn \l__xeCJK_bound_dim
3586   { \_xeCJK_use_punct_dim:nn { bound_width } #1 #2 }
3587   \dim_compare:nNf \l__xeCJK_bound_dim > \c_zero_dim
3588   {
3589     \dim_compare:nNf \l__xeCJK_last_bound_dim > \c_zero_dim
3590     {
3591       \dim_set:Nn \l__xeCJK_last_bound_dim
3592       {
3593         - \l__xeCJK_last_bound_dim *
3594         \dim_ratio:nn { \c_max_dim } { 1pt }
3595       }
3596     }
3597     \_xeCJK_punct_bound_kern_ratio_aux:N #2
3598   }
3599 }
3600 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_ratio_aux:N #1
3601 {
3602   \skip_set:Nn \l__xeCJK_punct_kern_skip
3603   {
3604     \l__xeCJK_punct_kern_skip *
3605     \dim_ratio:nn
3606     {
3607       \l__xeCJK_last_bound_dim
3608       + \_xeCJK_use_punct_dim:nn { bound } \c_xeCJK_left_tl #1
3609     }
3610   }

```

```

3610         { \l__xeCJK_bound_dim }
3611     }
3612 }

\__xeCJK_nobreak_hskip:N 3613 \cs_new_protected:Npn \__xeCJK_nobreak_hskip:N
\__xeCJK_nobreak_hskip:n 3614 { \xeCJK_no_break: \skip_horizontal:N }
\__xeCJK_punct_bound_kern:N 3615 \cs_new_protected:Npn \__xeCJK_nobreak_hskip:n
\__xeCJK_punct_bound_breakable_kern:N 3616 { \xeCJK_no_break: \skip_horizontal:n }
3617 \cs_new_eq:NN \__xeCJK_punct_bound_kern:N \__xeCJK_nobreak_hskip:N
3618 \cs_new_protected:Npn \__xeCJK_punct_bound_breakable_kern:N
3619 {
3620     \tl_if_eq:NNTF \l__xeCJK_aligni_tl \c__xeCJK_right_tl
3621     {
3622         \tl_if_eq:NNTF \l__xeCJK_alignii_tl \c__xeCJK_left_tl
3623         { \skip_horizontal:N }
3624         { \__xeCJK_nobreak_hskip:N }
3625     }
3626     { \__xeCJK_nobreak_hskip:N }
3627 }
3628 \cs_new_protected:Npn \__xeCJK_punct_bound_kern_aux:NNN #1#2
3629 {
3630     \str_if_eq:nnTF {#1} {#2}
3631     { \__xeCJK_nobreak_hskip:N }
3632     {
3633         \__xeCJK_punct_if_long:NTF #1
3634         { \skip_horizontal:N }
3635         {
3636             \__xeCJK_punct_if_long:NTF #2
3637             { \skip_horizontal:N }
3638             { \__xeCJK_punct_bound_kern:N }
3639         }
3640     }
3641 }

3642 \clist_map_inline:nn { CJK , FullLeft , FullRight }
3643 {
3644     \clist_map_inline:nn { FullLeft , FullRight }
3645     {
3646         \xeCJK_inter_class_toks:nne {#1} {##1}
3647         { \exp_not:c { xeCJK_#1_and_##1:N } }
3648     }
3649 }

```

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

```

3650 \cs_new_protected:Npn \__xeCJK_punct_bound_rule:NN #1#2
3651 {
3652     \tex_vrule:D
3653     width - \__xeCJK_use_punct_dim:nNN { bound } #1 #2 ~
3654     depth \c_zero_dim
3655     height \c_zero_dim \scan_stop:
3656 }

```

__xeCJK_punct_rule:NN 用于减少标点符号的左/右空白。

```

3657 \cs_new_protected:Npn \__xeCJK_punct_rule:NN #1#2
3658 {
3659     \tex_vrule:D
3660     width \__xeCJK_use_punct_dim:nNN { rule } #1 #2 ~
3661     depth \c_zero_dim
3662     height \c_zero_dim \scan_stop:
3663 }

```

__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。

```

3664 \cs_new_protected:Npn \__xeCJK_punct_glue:NN #1#2
3665 { \__xeCJK_punct_hskip:n { \__xeCJK_use_dim_or_skip:nNN { glue } #1 #2 } }
3666 \cs_new_eq:NN \__xeCJK_punct_hskip:n \skip_horizontal:n

```

\xeCJK_punct_kern:NN 相邻两个标点之间的间距。长标点与其他标点之间允许折行，但断点两侧的禁则必须同时满足，由 _xeCJK_punct_kern_break:NN 判断。

```

3667 \cs_new_protected:Npn \_xeCJK_punct_kern:NN #1#2
3668 {
3669   \str_if_eq:eeTF {#1} {#2}
3670   { \_xeCJK_punct_nobreak_kern:NN }
3671   {
3672     \bool_lazy_or:nnTF
3673     { \_xeCJK_punct_if_long_p:N #1 }
3674     { \_xeCJK_punct_if_long_p:N #2 }
3675     { \_xeCJK_punct_kern_break:NN #1 #2 }
3676     { \_xeCJK_punct_nobreak_kern:NN }
3677   }
3678   #1 #2
3679 }
3680 \cs_new_eq:NN \xeCJK_punct_kern:NN \_xeCJK_punct_kern:NN

```

_xeCJK_punct_kern_break:NN 长标点与其他标点相邻时，判断二者之间的断点是否满足两侧禁则，并留下相应的间距函数（吃掉参数后，后续的 #1 #2 交由留下的函数处理）：

- 断点之前(#1)必须是全角右标点或长标点，即全角左标点不得悬于行尾；
- 断点之后(#2)必须是全角左标点，或者不属于 NoBreakLongPunct 的长标点，即全角右标点和省略号等不得落于行首。

注意 #1 来自 \g__xeCJK_last_punct_tl，参与字符类判断前需要先展开为字符记号。

```

3681 \cs_new_protected:Npn \_xeCJK_punct_kern_break:NN #1#2
3682 {
3683   \exp_after:wN \_xeCJK_punct_if_right:NTF #1
3684   {
3685     \_xeCJK_punct_if_right:NTF #2
3686     {
3687       \bool_lazy_and:nnTF
3688       { \_xeCJK_punct_if_long_p:N #2 }
3689       { ! \_xeCJK_punct_if_nobreak_long_p:N #2 }
3690       { \_xeCJK_punct_breakable_kern:NN }
3691       { \_xeCJK_punct_nobreak_kern:NN }
3692     }
3693     { \_xeCJK_punct_breakable_kern:NN }
3694   }
3695   { \_xeCJK_punct_nobreak_kern:NN }
3696 }

```

_xeCJK_punct_nobreak_kern:NN 3697 \cs_new_protected:Npn _xeCJK_punct_nobreak_kern:NN #1#2
3698 { _xeCJK_nobreak_hskip:n { _xeCJK_use_dim_or_skip:nNN { kern } #1 #2 } }

_xeCJK_punct_breakable_kern:NN 3699 \cs_new_protected:Npn _xeCJK_punct_breakable_kern:NN #1#2
3700 {
3701 \exp_after:wN _xeCJK_punct_if_right:NT #1
3702 { _xeCJK_punct_rule:NN \c__xeCJK_right_tl #1 }
3703 _xeCJK_punct_breakable_kern:n
3704 { _xeCJK_use_dim_or_skip:nNN { bound_kern } #1 #2 }
3705 _xeCJK_punct_if_right:NF #2
3706 { _xeCJK_punct_rule:NN \c__xeCJK_left_tl #2 }
3707 }
3708 \cs_new_eq:NN _xeCJK_punct_breakable_kern:n \skip_horizontal:n

\g__xeCJK_last_punct_tl 用于记录当前的标点符号。

```

3709 \tl_new:N \g__xeCJK_last_punct_tl

```

```

\xeCJK_FullLeft_and_CJK: 3710 \cs_new_protected:Npn \xeCJK_FullLeft_and_CJK:
3711 {
3712   \_xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3713   {
3714     \_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3715     \xeCJK_no_break:

```

```

3716     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3717   }
3718   { }
3719   \__xeCJK_select_font:
3720 }

```

\xeCJK_FullLeft_and_Default: __xeCJK_nobreak_zero_glue: 用于确保 FullLeft 类后的西文单词可以断词。

```

3721 \cs_new_protected:Npn \xeCJK_FullLeft_and_Default:
3722 {
3723   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3724   {
3725     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3726     \xeCJK_class_group_end: \xeCJK_no_break:
3727     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3728   }
3729   {
3730     \xeCJK_class_group_end:
3731     \__xeCJK_nobreak_zero_glue:
3732   }
3733 }
3734 \cs_new_protected:Npn \__xeCJK_nobreak_zero_glue:
3735 {
3736   \tex_penalty:D \c__xeCJK_nobreak_penalty_int
3737   \skip_horizontal:N \c_zero_skip
3738 }
3739 \cs_new_protected:Npn \__xeCJK_zero_glue:
3740 { \skip_horizontal:N \c_zero_skip }

```

```

\xeCJK_FullRight_and_CJK: 3741 \cs_new_protected:Npn \xeCJK_FullRight_and_CJK:
3742 {
3743   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3744   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3745   \__xeCJK_select_font:
3746   \CJKglue
3747 }

```

```

\xeCJK_FullRight_and_Default: 3748 \cs_new_protected:Npn \xeCJK_FullRight_and_Default:
3749 {
3750   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3751   \xeCJK_class_group_end:
3752   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3753 }

```

```

\xeCJK_Default_and_FullLeft:nN 3754 \cs_new_protected:Npn \xeCJK_Default_and_FullLeft:nN #1#2
3755 {
3756   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #2
3757   \__xeCJK_Default_and_FullLeft_glue:N #2
3758   \xeCJK_class_group_begin:
3759   \xeCJK_select_punct_font:
3760   \xeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
3761   \xeCJK_clear_Boundary_and_CJK_toks:
3762   \tl_gset:Nn \g__xeCJK_last_punct_tl {#2}
3763   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #2
3764   \xeCJK_fallback_punct_symbol:NN
3765   \CJKpunctsymbol #2
3766 }
3767 \cs_new_protected:Npn \__xeCJK_Default_and_FullLeft_glue:N #1
3768 { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1 }

```

```

\xeCJK_CJK_and_FullLeft:N 3769 \cs_new_protected:Npn \xeCJK_CJK_and_FullLeft:N #1
3770 {
3771   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
3772   \__xeCJK_CJK_and_FullLeft_glue:N #1
3773   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
3774   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
3775   \__xeCJK_select_punct_font:
3776   \xeCJK_fallback_punct_symbol:NN
3777   \CJKpunctsymbol #1

```

```

3778 }
3779 \cs_new_protected:Npn \__xeCJK_CJK_and_FullLeft_glue:N #1
3780 {
3781   \CJKglue
3782   \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
3783 }
\xeCJK_Boundary_and_FullLeft:N 3784 \cs_new_protected:Npn \xeCJK_Boundary_and_FullLeft:N #1
3785 {
3786   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
3787   \__xeCJK_Boundary_and_FullLeft_glue:N #1
3788   \xeCJK_class_group_begin:
3789   \xeCJK_select_punct_font:
3790   \xeCJK_clear_Boundary_and_CJK_toks:
3791   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
3792   \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
3793   \xeCJK_fallback_punct_symbol:NN
3794   \CJKpunctsymbol #1
3795 }

```

根据 `\etex_lastnodetype:D` 的值进行分别处理。

```

3796 \cs_new_protected:Npn \__xeCJK_Boundary_and_FullLeft_glue:N #1
3797 {
3798   \tl_set_eq:NN \l__xeCJK_alignii_tl \c__xeCJK_left_tl
3799   \group_begin: \exp_args:NNc \group_end: \cs_if_exist_use:NTF
3800   { \__xeCJK_bound_type_ \int_use:N \etex_lastnodetype:D _glue:Nn }
3801   {#1}
3802   { \use:n }
3803   { \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1 }
3804 }
3805 \tl_new:N \c__xeCJK_alignii_tl

```

`\etex_lastnodetype:D` 为 `-1` 表示 empty list, 常出现在盒子的起始位置, 在段落前使用 `\noindent` 就是这种情况。

```

3806 \cs_new_protected:cpn { \__xeCJK_bound_type_ -1 _glue:Nn } #1#2
3807 { \__xeCJK_zero_glue: }

```

`1` 表示 hlist node, 在这里用来判断是否位于段首。基于正常情况下, $\TeX$ 会在段落开头插入宽度为 `\parindent` 的水平盒子用于缩进。

```

3808 \cs_new_protected:cpn { \__xeCJK_bound_type_ 1 _glue:Nn } #1
3809 {
3810   \int_do_while:nNnn \etex_lastnodetype:D = \c__xeCJK_hlist_node
3811   { \__xeCJK_bound_hbox_auxi: }
3812   \__xeCJK_if_last_none:TF
3813   {
3814     \dim_case:nnF { \box_wd:N \l__xeCJK_indent_box }
3815     {
3816       { \etex_parindent:D } { \__xeCJK_bound_hbox_auxii:nn }
3817       { \c_zero_dim } { \use_i:nn }
3818     }
3819     { \use:nn }
3820   }
3821   { \use:nn }
3822   { \hbox_unpack_drop:N \l__xeCJK_indent_box }
3823 }
3824 \cs_new_protected:Npn \__xeCJK_bound_hbox_auxi:
3825 {
3826   \box_set_to_last:N \l__xeCJK_tmp_box
3827   \hbox_set:Nn \l__xeCJK_indent_box
3828   {
3829     \box_use:N \l__xeCJK_tmp_box
3830     \hbox_unpack:N \l__xeCJK_indent_box
3831   }
3832 }
3833 \cs_new_protected:Npn \__xeCJK_bound_hbox_auxii:nn

```

```

3834 {
3835   \dim_compare:nNnTF
3836     { \box_ht:N \l__xeCJK_tmp_box } = \c_zero_dim
3837     { \use_i:nn }
3838     { \use:nn }
3839 }
3840 \box_new:N \l__xeCJK_indent_box

```

11 表示 glue node, 这里判断的目的是当全角左标点出现在 $\text{\LaTeX}$ 表格的非 p 列行首时, 能够对齐到单元格的边界。判断基于标准 $\text{\LaTeX}$ 表格的列格式 ($\text{\@tabclassz}$) 定义中, 在 l 列和 r 列前为了防止 $\text{\tabcolsep}$ 被无意 $\text{\unskip}$ 掉, 都加了 $\text{\hskip1sp}$, 而 c 列前则有 $\text{\hfil}$ 。 $\text{\enumitem}$ 宏包修改了 description 环境中使用的 $\text{\item(\enit@postlabel@i)}$, 在这里起到影响作用的是 $\text{\penalty\z@ \hskip\labelsep}$ 。

```

3841 \cs_new_protected:cpn { __xeCJK_bound_type_ 11 _glue:Nn } #1#2
3842 {
3843   \skip_if_finite:nTF { \tex_lastskip:D }
3844     { \__xeCJK_bound_glue_auxi:Nn #1 {#2} }
3845     { \__xeCJK_zero_glue: }
3846 }
3847 \cs_new_protected:Npn \__xeCJK_bound_glue_auxi:Nn #1#2
3848 {
3849   \__xeCJK_if_last_punct_glue:TF
3850     { \xeCJK_punct_bound_kern:N #1 }
3851     { \__xeCJK_bound_glue_auxii:n {#2} }
3852 }
3853 \cs_new_protected:Npn \__xeCJK_bound_glue_auxii:n #1
3854 {
3855   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
3856   \skip_if_eq:nnTF { \l__xeCJK_last_skip } { 1sp }
3857     { \__xeCJK_zero_glue: }
3858     {
3859       \skip_if_eq:nnTF { \l__xeCJK_last_skip } { \labelsep }
3860         {
3861           \tex_unskip:D
3862           \__xeCJK_if_last_penalty:TF
3863             {
3864               \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
3865                 { \skip_horizontal:N \l__xeCJK_last_skip }
3866                 { \skip_horizontal:N \l__xeCJK_last_skip #1 }
3867             }
3868             { \skip_horizontal:N \l__xeCJK_last_skip #1 }
3869         }
3870         {#1}
3871     }
3872 }

```

12 表示 kern node, 用于判断之前的字符是否是 CJK 类, 如果是, 则插入 $\text{\CJKglue}$ 。

```

3873 \cs_new_protected:cpn { __xeCJK_bound_type_ 12 _glue:Nn } #1#2
3874 {
3875   \xeCJK_if_last_node:nF { CJK }
3876     { \xeCJK_if_last_node:nF { CJK-space } { \use_none:nn } }
3877   \xeCJK_remove_node: \CJKglue
3878   #2
3879 }

```

13 表示 penalty node, 这里判断的目的是全角左标点出现在 $\text{\LaTeX}$ 列表环境的 $\text{\item}$ 后面时, 能对齐到边界。判断基于 $\text{\item}$ 的内部定义 $\text{\@item}$ 对 $\text{\everypar}$ 进行了修改, 在这里起到影响作用的是 $\text{\box\@labels \penalty\z@}$ 。以上判断都比较粗略, 暂时也没有想起更好的办法。

```

3880 \cs_new_protected:cpn { __xeCJK_bound_type_ 13 _glue:Nn } #1#2
3881 {
3882   \__xeCJK_if_last_punct_penalty:TF
3883     { \xeCJK_punct_bound_kern:N #1 }
3884     {

```

```

3885         \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
3886         {
3887             \tex_unpenalty:D
3888             \__xeCJK_if_last_hlist:TF
3889             { \tex_penalty:D \c_zero_int }
3890             { \tex_penalty:D \c_zero_int #2 }
3891         }
3892     {#2}
3893 }
3894 }

\cs_new_protected:Npn \xeCJK_Default_and_FullRight:nN #1#2
3895 {
3896     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #2
3897     \__xeCJK_Default_and_FullRight_glue:N #2
3898     \xeCJK_class_group_begin:
3899     \xeCJK_select_punct_font:
3900     \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
3901     \xeCJK_clear_Boundary_and_CJK_toks:
3902     \tl_gset:Nn \g__xeCJK_last_punct_tl {#2}
3903     \xeCJK_FullRight_symbol:N #2
3904 }

\cs_new_protected:Npn \xeCJK_Boundary_and_FullRight:N #1
3906 {
3907     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
3908     \xeCJK_if_last_punct:TF
3909     {
3910         \tl_set_eq:NN \l__xeCJK_alignii_tl \c__xeCJK_right_tl
3911         \xeCJK_punct_bound_kern:N
3912     }
3913     { \__xeCJK_Default_and_FullRight_glue:N }
3914     #1
3915     \xeCJK_class_group_begin:
3916     \xeCJK_select_punct_font:
3917     \xeCJK_clear_Boundary_and_CJK_toks:
3918     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
3919     \xeCJK_FullRight_symbol:N #1
3920 }

\cs_new_protected:Npn \xeCJK_CJK_and_FullRight:N #1
3922 {
3923     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
3924     \__xeCJK_CJK_and_FullRight_glue:N #1
3925     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
3926     \__xeCJK_select_punct_font:
3927     \xeCJK_FullRight_symbol:N #1
3928 }

\cs_new_protected:Npn \xeCJK_if_last_punct:TF
3930 {
3931     \bool_set_false:N \l__xeCJK_last_penalty_bool
3932     \__xeCJK_if_last_glue:TF
3933     { \__xeCJK_if_last_punct_glue:TF }
3934     {
3935         \__xeCJK_if_last_penalty:TF
3936         { \__xeCJK_if_last_punct_penalty:TF }
3937         { \use_ii:nn }
3938     }
3939 }

\cs_new_protected:Npn \__xeCJK_if_last_punct_glue:TF
3941 {
3942     \prop_get:NoNTF \g__xeCJK_punct_skip_prop
3943     { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
3944     { \__xeCJK_if_last_punct_glue_auxi:TF }
3945     { \__xeCJK_if_last_punct_glue_auxii:TF }
3946 }

```

`\xeCJK_if_last_punct:TF` 判断之前是否是一个标点符号。

```

3947 }
3948 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxi:TF
3949 {
3950   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
3951   \tex_unskip:D
3952   \int_compare:nNnTF \tex_lastpenalty:D = \c__xeCJK_nobreak_penalty_int
3953   { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
3954   {
3955     \xeCJK_if_last_node:TF
3956     { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
3957     { \use:n }
3958   }
3959   { \skip_horizontal:N \l__xeCJK_last_skip \use_ii:nn }
3960 }
3961 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxii:TF
3962 {
3963   \group_begin:
3964   \g__xeCJK_space_factor_int \tex_spacefactor:D
3965   \skip_if_eq:nnTF { \tex_lastskip:D } { \c__xeCJK_space_skip_tl }
3966   { \group_end: \__xeCJK_if_last_punct_glue_auxiii:TF }
3967   { \group_end: \use_ii:nn }
3968 }
3969 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxiii:TF
3970 {
3971   \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
3972   \tex_unskip:D
3973   \__xeCJK_if_last_glue:TF
3974   {
3975     \prop_get:NnTF \g__xeCJK_punct_skip_prop
3976     { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
3977     { \__xeCJK_if_last_punct_glue_auxi:TF { \use_i:nn } }
3978     { \use:n }
3979   }
3980   { \use:n }
3981   { \skip_horizontal:N \l__xeCJK_tmp_skip \use_ii:nn }
3982 }
3983 \cs_new_protected:Npn \__xeCJK_if_last_punct_penalty:TF
3984 {
3985   \int_set_eq:NN \l__xeCJK_last_penalty_int \tex_lastpenalty:D
3986   \tex_unpenalty:D
3987   \bool_set_true:N \l__xeCJK_last_penalty_bool
3988   \__xeCJK_if_last_glue:TF
3989   { \__xeCJK_if_last_punct_glue:TF { \use_i:nn } }
3990   { \use:n }
3991   { \__xeCJK_last_punct_penalty_false:nn }
3992 }
3993 \cs_new_protected:Npn \__xeCJK_last_punct_penalty_false:nn #1#2
3994 {
3995   \bool_set_false:N \l__xeCJK_last_penalty_bool
3996   \tex_penalty:D \l__xeCJK_last_penalty_int
3997   #2
3998 }
3999 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxi:TF
4000 {
4001   \tex_unpenalty:D
4002   \bool_if:NF \l__xeCJK_last_penalty_bool
4003   {
4004     \bool_set_true:N \l__xeCJK_last_penalty_bool
4005     \int_set_eq:NN \l__xeCJK_last_penalty_int \c__xeCJK_nobreak_penalty_int
4006   }
4007   \xeCJK_if_last_node:TF
4008   { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
4009   { \use:n }
4010   { \xeCJK_no_break: \use_ii:nn }
4011 }
4012 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxii:TF
4013 {

```

```

4014 \dim_compare:nNnTF \l__xeCJK_last_kern_dim > \c_zero_dim
4015 { \__xeCJK_if_last_punct_auxiii:TF }
4016 { \__xeCJK_make_node:N \l__xeCJK_last_kern_dim \use_ii:nn }
4017 }
4018 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiii:TF
4019 {
4020 \int_case:nnTF { \tex_XeTeXcharclass:D \l__xeCJK_last_kern_dim }
4021 {
4022 { \xeCJK_class_num:n { FullRight } }
4023 { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_right_tl }
4024 { \xeCJK_class_num:n { FullLeft } }
4025 { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_left_tl }
4026 }
4027 { \__xeCJK_if_last_punct_auxiv:TF }
4028 { \use_ii:nn }
4029 }
4030 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiv:TF
4031 {
4032 \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
4033 \xeCJK_if_last_node:TF
4034 {
4035 \tl_gset:Ne \g__xeCJK_last_punct_tl
4036 { \tex_Uchar:D \l__xeCJK_tmp_dim }
4037 \dim_set_eq:NN \l__xeCJK_last_bound_dim \l__xeCJK_last_kern_dim
4038 \use_i:nn
4039 }
4040 { \__xeCJK_make_node:N \l__xeCJK_tmp_dim \use_ii:nn }
4041 }
4042 \tl_new:N \l__xeCJK_aligni_tl
4043 \tl_new:N \l__xeCJK_alignii_tl
4044 \int_new:N \l__xeCJK_last_penalty_int
4045 \dim_new:N \l__xeCJK_last_bound_dim
4046 \bool_new:N \l__xeCJK_last_penalty_bool

\xeCJK_if_last_node:TF 4047 \cs_new_protected:Npn \xeCJK_if_last_node:TF #1#2
4048 {
4049 \__xeCJK_if_last_kern:TF
4050 {
4051 \dim_set_eq:NN \l__xeCJK_last_kern_dim \tex_lastkern:D
4052 \tex_unkern:D
4053 \__xeCJK_if_last_kern:TF
4054 {
4055 \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_last_kern_dim }
4056 { \tex_unkern:D #1 }
4057 { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
4058 }
4059 { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
4060 }
4061 {#2}
4062 }
4063 \dim_new:N \l__xeCJK_last_kern_dim

\__xeCJK_CJK_and_FullRight_glue:N 4064 \cs_new_protected:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
\__xeCJK_Default_and_FullRight_glue:N 4065 {
4066 \__xeCJK_punct_if_long:NTF #1
4067 {
4068 \__xeCJK_punct_if_nobreak_long:NTF #1
4069 { \xeCJK_no_break: }
4070 { \xeCJK_allow_break: }
4071 }
4072 { \xeCJK_no_break: }
4073 \__xeCJK_punct_if_middle:NT #1
4074 {
4075 \CJKglue
4076 \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
4077 \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
4078 }

```

```

4079 }
4080 \cs_new_protected:Npn \__xeCJK_Default_and_FullRight_glue:N #1
4081 {
4082   \__xeCJK_punct_if_long:NTF #1
4083   {
4084     \__xeCJK_punct_if_nobreak_long:NTF #1
4085     { \xeCJK_no_break: }
4086     { \xeCJK_allow_break: }
4087   }
4088   { \xeCJK_no_break: }
4089   \__xeCJK_punct_if_middle:NT #1
4090   {
4091     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
4092     \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
4093   }
4094 }

\xeCJK_FullLeft_and_FullLeft:N 4095 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullLeft:N #1
4096 {
4097   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
4098   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4099   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4100   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4101   \xeCJK_fallback_punct_symbol:NN
4102   \CJKpunctsymbol #1
4103 }

\xeCJK_FullLeft_and_FullRight:N 4104 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullRight:N #1
4105 {
4106   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4107   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4108   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4109   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4110   \xeCJK_FullRight_symbol:N #1
4111 }

\xeCJK_FullRight_and_FullLeft:N 4112 \cs_new_protected:Npn \xeCJK_FullRight_and_FullLeft:N #1
4113 {
4114   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
4115   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4116   \xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4117   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4118   \xeCJK_fallback_punct_symbol:NN
4119   \CJKpunctsymbol #1
4120 }

\xeCJK_FullRight_and_FullRight:N 4121 \cs_new_protected:Npn \xeCJK_FullRight_and_FullRight:N #1
4122 {
4123   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4124   \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4125   \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4126   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4127   \xeCJK_FullRight_symbol:N #1
4128 }

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```

4129 \keys_define:nn { xeCJK / options }
4130 {
4131   CheckFullRight .choice: ,
4132   CheckFullRight / true .code:n =
4133   {
4134     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
4135     {
4136       \cs_set_eq:NN \__xeCJK_save_FullRight_check:
4137       \xeCJK_FullRight_and_Boundary:

```

```

4138         \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N
4139         \xeCJK_FullRight_symbol:N
4140         \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
4141         \xeCJK_check_FullRight:
4142         \cs_set_eq:NN \xeCJK_FullRight_symbol:N
4143         \xeCJK_check_FullRight_symbol:Nw
4144     }
4145 },
4146 CheckFullRight / false .code:n =
4147 {
4148     \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
4149     {
4150         \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
4151         \__xeCJK_save_FullRight_check:
4152         \cs_set_eq:NN \xeCJK_FullRight_symbol:N
4153         \__xeCJK_save_FullRight_symbol:N
4154     }
4155 },
4156 CheckFullRight .default:n = { true }
4157 }

experiment/halfright-prebreakpenalty 4158 \keys_define:nn { xeCJK / boundary-register }
experiment/punct-measure-fix          4159 {
experiment/boundary-register          4160     command .code:n =
4161     { \__xeCJK_boundary_user_register_command:n {#1} } ,
4162     command .value_required:n = true ,
4163     strategy .code:n =
4164     { \__xeCJK_boundary_user_register_strategy:n {#1} } ,
4165     strategy .value_required:n = true ,
4166     mode .code:n =
4167     { \__xeCJK_boundary_user_register_mode:n {#1} } ,
4168     mode .value_required:n = true
4169 }
4170 \keys_define:nn { xeCJK / options / experiment }
4171 {
4172     halfright-prebreakpenalty .bool_set:N = \l__xeCJK_halfright_nobreak_bool ,
4173     halfright-prebreakpenalty .default:n = true ,
4174     halfright-prebreakpenalty .initial:n = false ,
4175     punct-measure-fix .bool_gset:N = \g__xeCJK_punct_measure_fix_bool ,
4176     punct-measure-fix .default:n = true ,
4177     punct-measure-fix .initial:n = false ,
4178     boundary-register .code:n =
4179     { \__xeCJK_boundary_user_register:n {#1} } ,
4180     boundary-register .value_required:n = true
4181 }

\xeCJK_FullRight_symbol:N 4182 \cs_new_protected:Npn \xeCJK_FullRight_symbol:N
4183 {
4184     \xeCJK_fallback_punct_symbol:NN
4185     \CJKpunctsymbol
4186 }

\xeCJK_check_FullRight: 4187 \cs_new_protected:Npn \xeCJK_check_FullRight:
4188 {
4189     \xeCJK_get_punct_bounds:No \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4190     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4191     \group_align_safe_begin:
4192     \token_case_meaning:NoTF \l_peek_token
4193     { \l__xeCJK_no_break_cs_case_tl }
4194     {
4195         \group_align_safe_end:
4196         \xeCJK_no_break:
4197         \group_insert_after:N \xeCJK_no_break:
4198     }
4199     { \group_align_safe_end: }
4200     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
4201     \xeCJK_class_group_end:
4202     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl

```

```

4203     \__xeCJK_punct_boundary_guard:
4204   }
4205   \prg_generate_conditional_variant:Nnn \token_case_meaning:Nn { No } { TF , F }

\XeCJK_check_FullRight_symbol:Nw 4206 \cs_new_protected:Npn \xeCJK_check_FullRight_symbol:Nw #1
4207 { \peek_remove_spaces:n { \__xeCJK_save_FullRight_symbol:N #1 } }

\XeCJK_cs_case_keys_define:nNNnn 4208 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
4209 {
4210   \tl_new:N #2
4211   \seq_new:N #3
4212   \keys_define:nn { xeCJK / options }
4213   {
4214     #1 .code:n =
4215     {
4216       \seq_set_split:Nnn #3 { } {##1}
4217       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
4218     } ,
4219     #1+ .code:n =
4220     {
4221       \tl_map_inline:nn {##1}
4222       { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
4223       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
4224     } ,
4225     #1- .code:n =
4226     {
4227       \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {####1} }
4228       \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
4229     }
4230   }
4231 }
4232 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
4233 {
4234   \tl_clear:N #1
4235   \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
4236   #4
4237 }

```

NoBreakCS 设置不能在全角右标点之后断行的控制序列。

```

4238 \xeCJK_cs_case_keys_define:nNNnn { NoBreakCS }
4239 \l__xeCJK_no_break_cs_case_tl \l__xeCJK_no_break_cs_seq { } { }

```

\xeCJKnobreak 为保险起见, 我们在这里用了一个循环。

```

4240 \NewDocumentCommand \xeCJKnobreak { }
4241 {
4242   \bool_set_true:N \l__xeCJK_tmp_bool
4243   \int_while_do:nNnn \tex_lastnodetype:D = \c__xeCJK_glue_node
4244   {
4245     \bool_if:NTF \l__xeCJK_tmp_bool
4246     {
4247       \bool_set_false:N \l__xeCJK_tmp_bool
4248       \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
4249     }
4250     { \skip_add:Nn \l__xeCJK_last_skip \tex_lastskip:D }
4251     \tex_unskip:D
4252   }
4253   \xeCJK_if_last_node:TF
4254   {
4255     \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
4256     \xeCJK_if_last_node:TF
4257     {
4258       \__xeCJK_if_last_glue:T
4259       {
4260         \exp_args:NNNo \tex_unskip:D \xeCJK_no_break:
4261         \skip_horizontal:n { \skip_use:N \tex_lastskip:D }
4262       }
4263       \__xeCJK_make_node:N \l__xeCJK_last_kern_dim

```

```

4264     }
4265     { }
4266     \__xeCJK_make_node:N \l__xeCJK_tmp_dim
4267   }
4268   { }
4269   \xeCJK_no_break:
4270   \bool_if:NF \l__xeCJK_tmp_bool
4271     { \skip_horizontal:N \l__xeCJK_last_skip }
4272 }

```

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```

4273 \keys_define:nn { xeCJK / options }
4274 {
4275   CheckSingle .choice: ,
4276   CheckSingle / true .code:n =
4277     {
4278       \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4279       {
4280         \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
4281         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4282       }
4283     } ,
4284   CheckSingle / false .code:n =
4285     {
4286       \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4287       { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
4288     } ,
4289   CheckSingle .default:n = { true } ,
4290   CJKchecksingle .meta:n = { CheckSingle = true }
4291 }

```

WidowPenalty 设置段末汉字的 penalty, 默认值是 10 000。

```

4292 \keys_define:nn { xeCJK / options }
4293 {
4294   WidowPenalty .int_set:N = \l__xeCJK_widow_penalty_int ,
4295   WidowPenalty .default:n = { 10 000 }
4296 }

```

\xeCJK_widow_penalty: 预防段末孤字而插入的 penalty, 值为 \l__xeCJK_widow_penalty_int。

```

4297 \cs_new_protected:Npn \xeCJK_widow_penalty:
4298 { \tex_penalty:D \l__xeCJK_widow_penalty_int }

```

```

\__xeCJK_check_single:Nw 4299 \cs_new_protected:Npn \xeCJK_check_single:Nw #1
\__xeCJK_check_single_end:N 4300 {
4301   \group_align_safe_begin:
4302   \peek_catcode:NTF \c_catcode_letter_token
4303     { \xeCJK_check_single:NNw #1 }
4304     {
4305       \token_if_other:NNTF \l_peek_token
4306         { \xeCJK_check_single:NNw }
4307         { \__xeCJK_check_single_end:N }
4308       #1
4309     }
4310 }
4311 \cs_new_protected:Npn \__xeCJK_check_single_end:N
4312 {
4313   \group_align_safe_end:
4314   \__xeCJK_check_single_save:N
4315 }

```

使用 \group_align_safe_begin: 和 \group_align_safe_end: 是为了防止在表格里面报错。

```

\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。
\_XeCJK_check_single_aux:nnw
4316 \cs_new_protected:Npn \XeCJK_check_single:NNw #1#2
4317 {
4318   \XeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
4319   {
4320     \bool_if:NTF \l__XeCJK_peek_ignore_spaces_bool
4321     {
4322       \bool_if:NTF \l__XeCJK_reserve_space_bool
4323       { \__XeCJK_check_single_end:N #1 #2 ~ }
4324       { \__XeCJK_check_single_space:NN #1#2 }
4325     }
4326     { \__XeCJK_check_single_end:N #1 #2 }
4327   }
4328   {
4329     \token_if_other:NTF \l_peek_token
4330     {
4331       \bool_if:NTF \l__XeCJK_peek_ignore_spaces_bool
4332       { \__XeCJK_check_single_space:NN }
4333       { \__XeCJK_check_single_end:N }
4334     }
4335     {
4336       \bool_if:NTF \l__XeCJK_peek_ignore_spaces_bool
4337       { \__XeCJK_check_single_aux:nnw { ~ } }
4338       { \__XeCJK_check_single_aux:nnw { } }
4339     }
4340     #1 #2
4341   }
4342 }
4343 \cs_new_protected:Npn \__XeCJK_check_single_aux:nnw #1#2#3
4344 {
4345   \token_if_cs:NTF \l_peek_token
4346   { \XeCJK_check_single_cs:NNn }
4347   { \XeCJK_check_single_end:NNnw }
4348   #2 #3 {#1}
4349 }

\XeCJK_check_single_end:NNnw 4350 \cs_new_protected:Npn \__XeCJK_check_single_end_aux:NNn #1#2#3
\_XeCJK_check_single_end_aux:NNn 4351 { \__XeCJK_check_single_end:N #1 #2 #3 }
\_XeCJK_check_single_end_equation:NNnw 4352 \cs_new_eq:NN \XeCJK_check_single_end:NNnw \__XeCJK_check_single_end_aux:NNn
4353 \cs_new_protected:Npn \__XeCJK_check_single_end_equation:NNnw
4354 {
4355   \token_if_math_toggle:NTF \l_peek_token
4356   { \XeCJK_check_single_equation:NNnw }
4357   { \__XeCJK_check_single_end_aux:NNn }
4358 }

PlainEquation 4359 \keys_define:nn { XeCJK / options }
4360 {
4361   PlainEquation .choice: ,
4362   PlainEquation / true .code:n =
4363   {
4364     \cs_set_eq:NN \XeCJK_check_single_end:NNnw
4365     \__XeCJK_check_single_end_equation:NNnw
4366   } ,
4367   PlainEquation / false .code:n =
4368   {
4369     \cs_set_eq:NN \XeCJK_check_single_end:NNnw
4370     \__XeCJK_check_single_end_aux:NNn
4371   } ,
4372   PlainEquation .default:n = { true } ,
4373 }

\_XeCJK_check_single_space:NN 4374 \cs_new_protected:Npn \__XeCJK_check_single_space:NN #1#2
4375 {
4376   \XeCJK_if_CJK_class:NTF #2
4377   {
4378     \XeCJK_if_CJK_class:NTF \l_peek_token
4379     { \__XeCJK_check_single_end:N #1 #2 }

```

```

4380      { \_xeCJK_check_single_end:N #1 #2 ~ }
4381    }
4382    { \_xeCJK_check_single_end:N #1 #2 ~ }
4383  }

\_xeCJK_check_single_equation:NNnNw 4384 \cs_new_protected:Npn \xeCJK_check_single_equation:NNnNw #1#2#3#4
4385  {
4386    \peek_catcode:NTF \c_math_toggle_token
4387    {
4388      \xeCJK_widow_penalty: \_xeCJK_check_single_end:N #1
4389      \xeCJK_make_node:n { CJK-widow } #2 #4
4390    }
4391    { \_xeCJK_check_single_end:N #1 #2#3#4 }
4392  }

```

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

```

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

```

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

```

4393 \cs_new_protected:Npn \xeCJK_check_single_cs:NNn #1#2#3
4394  {
4395    \token_case_meaning:Nof \l_peek_token
4396    { \l_xeCJK_check_single_cs_case_tl }
4397    { \use_iii:nnn }
4398    { \xeCJK_check_single_env:nnNn }
4399    {
4400      \xeCJK_widow_penalty:
4401      \_xeCJK_check_single_end:N #1
4402      \xeCJK_make_node:n { CJK-widow } #2#3
4403    }
4404    { \_xeCJK_check_single_end:N #1 #2#3 }
4405  }
4406 \tl_new:N \l_xeCJK_check_single_cs_case_tl

\_xeCJK_check_single_env:nnNn 4407 \cs_new_protected:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
4408  {
4409    \str_case_e:noTF {#4}
4410    { \l_xeCJK_inline_env_case_tl }
4411    {#2}
4412    {#1}
4413    #3 {#4}
4414  }
4415 \prg_generate_conditional_variant:Nnn \str_case_e:nn { no } { TF }

NewLineCS 4416 \xeCJK_cs_case_keys_define:nNNnn { NewLineCS }
4417 \l_xeCJK_new_line_cs_case_tl \l_xeCJK_new_line_cs_seq
4418 { \use_ii:nnn }
4419 {
4420   \tl_concat:NNN \l_xeCJK_check_single_cs_case_tl
4421   \l_xeCJK_new_line_cs_case_tl \l_xeCJK_env_cs_case_tl
4422 }

EnvCS 4423 \xeCJK_cs_case_keys_define:nNNnn { EnvCS }
4424 \l_xeCJK_env_cs_case_tl \l_xeCJK_env_cs_seq
4425 { \use:n }
4426 {
4427   \tl_concat:NNN \l_xeCJK_check_single_cs_case_tl
4428   \l_xeCJK_new_line_cs_case_tl \l_xeCJK_env_cs_case_tl
4429 }

```

```

InlineEnv 4430 \keys_define:nn { xeCJK / options }
4431 {
4432   InlineEnv      .code:n =
4433   {
4434     \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
4435     \__xeCJK_update_inline_env_case_tl:
4436   } ,
4437   InlineEnv+      .code:n =
4438   {
4439     \clist_map_inline:nn {#1}
4440     {
4441       \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
4442       { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
4443     }
4444     \__xeCJK_update_inline_env_case_tl:
4445   } ,
4446   InlineEnv-      .code:n =
4447   {
4448     \clist_map_inline:nn {#1}
4449     { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
4450     \__xeCJK_update_inline_env_case_tl:
4451   }
4452 }
4453 \seq_new:N \l__xeCJK_inline_env_seq

\__xeCJK_update_inline_env_case_tl: 4454 \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
4455 {
4456   \tl_clear:N \l__xeCJK_inline_env_case_tl
4457   \seq_map_inline:Nn \l__xeCJK_inline_env_seq
4458   { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { } } }
4459 }
4460 \tl_new:N \l__xeCJK_inline_env_case_tl

```

5.9 增加 CJK 子分区

```
\g__xeCJK_CJK_sub_class_seq 4461 \seq_new:N \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。

```

4462 \NewDocumentCommand \xeCJKDeclareSubCJKBlock
4463 { s > { \TrimSpaces } m m }
4464 {
4465   \xeCJK_declare_sub_char_class:nen { CJK } {#2} {#3}
4466   \IfBooleanT {#1} { \xeCJKResetPunctClass }
4467 }
4468 \@onlypreamble \xeCJKDeclareSubCJKBlock

```

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

\xeCJKRestoreSubCJKBlock

```

4469 \bool_new:N \l__xeCJK_sub_cancel_bool
4470 \NewDocumentCommand \xeCJKCancelSubCJKBlock { s m }
4471 {
4472   \bool_if:NF \l__xeCJK_sub_cancel_bool
4473   {
4474     \bool_set_true:N \l__xeCJK_sub_cancel_bool
4475     \__xeCJK_sub_restore_or_cancel:e {#2}
4476     \IfBooleanT {#1} { \xeCJKResetPunctClass }
4477   }
4478 }
4479 \NewDocumentCommand \xeCJKRestoreSubCJKBlock { s m }
4480 {
4481   \bool_if:NT \l__xeCJK_sub_cancel_bool
4482   {
4483     \bool_set_false:N \l__xeCJK_sub_cancel_bool
4484     \__xeCJK_sub_restore_or_cancel:e {#2}
4485     \IfBooleanT {#1} { \xeCJKResetPunctClass }
4486   }
4487 }

```

```

\__xeCJK_sub_restore_or_cancel:n 4488 \cs_new_protected:Npn \__xeCJK_sub_restore_or_cancel:n #1
4489 {
4490   \clist_map_inline:nn {#1}
4491   {
4492     \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
4493     {
4494       \xeCJK_declare_char_class:nn
4495       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
4496       { \use:c { g__xeCJK_CJK/##1_range_clist } }
4497     }
4498     { \__xeCJK_error:ne { SubBlock-undefined } {##1} }
4499   }
4500 }
4501 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { e }
4502 \__xeCJK_msg_new:nn { SubBlock-undefined }
4503 {
4504   The~CJK~sub~block~`#1'~is~undefined.\\\
4505   Try~to~use~\token_to_str:N \xeCJKDeclareSubCJKBlock \
4506   to~declare~it.
4507 }

\xeCJK_declare_sub_char_class:nnn 4508 \cs_new_protected:Npn \xeCJK_declare_sub_char_class:nnn #1#2#3
4509 {
4510   \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
4511   {
4512     \xeCJK_new_class:n { #1/#2 }
4513     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
4514     \xeCJK_new_sub_key:n {#2}
4515   }
4516   \xeCJK_declare_char_class:nn { #1/#2 } {#3}
4517 }
4518 \cs_generate_variant:Nn \xeCJK_declare_sub_char_class:nnn { ne }

\__xeCJK_set_sub_class_toks:nn 4519 \cs_new_protected:Npn \__xeCJK_set_sub_class_toks:nn #1#2
4520 {
4521   \seq_map_inline:Nn \g__xeCJK_base_class_seq
4522   {
4523     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
4524     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
4525     \str_if_eq:nnTF {##1} { CJK }
4526     {
4527       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
4528       { \__xeCJK_switch_font:nn {#1} {#2} }
4529     }
4530     {
4531       \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
4532       { \xeCJK_fallback_symbol:NN }
4533       {
4534         \__xeCJK_switch_font:nn {#1} {#2}
4535         \xeCJK_fallback_symbol:NN
4536       }
4537     }
4538   }
4539   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
4540   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
4541   {
4542     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
4543     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
4544     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
4545     { \__xeCJK_switch_font:nn {#2} {##1} }
4546     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
4547     { \__xeCJK_switch_font:nn {##1} {#2} }
4548   }
4549   \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
4550   \__xeCJK_save_CJK_class:n { #1/#2 }
4551   \clist_map_inline:nn
4552   {
4553     CJK , FullLeft , FullRight , HangulJamo ,

```

```

4554     HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
4555   }
4556   {
4557     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
4558     { \_xeCJK_switch_font:nn {#2} {#1} }
4559   }
4560 }

```

5.10 标点处理

`\XeTeXglyphbounds` 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。

```

4561 \cs_if_exist:NF \tex_XeTeXglyphbounds:D
4562 {
4563   \_xeCJK_msg_new:nn { XeTeX-too-old }
4564   {
4565     \token_to_str:N \tex_XeTeXglyphbounds:D \ is~not~defined.\\
4566     CJK~punctuation~kerning~will~not~be~available.\\
4567     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
4568   }
4569   \_xeCJK_error:n { XeTeX-too-old }
4570   \AtEndOfPackage
4571   {
4572     \keys_define:nn { xeCJK / options }
4573     {
4574       PunctStyle .code:n =
4575       { \_xeCJK_error:ne { punct-style-unknown } {#1} }
4576     }
4577     \seq_gclear:N \g__xeCJK_punct_style_seq
4578     \_xeCJK_set_punct_style:n { plain }
4579   }
4580 }

```

`\xeCJKsetwidth` 手动设置参数中的标点符号的宽度。

```

4581 \NewDocumentCommand \xeCJKsetwidth { s m m }
4582 {
4583   \IfBooleanTF {#1}
4584   {
4585     \tl_map_inline:en {#2}
4586     {
4587       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
4588       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
4589       \tl_gset:cn { g__xeCJK_punct_bound_width/##1/tl } {#3}
4590     }
4591   }
4592   {
4593     \tl_map_inline:en {#2}
4594     {
4595       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
4596       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
4597       \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#3}
4598     }
4599   }
4600 }
4601 \onlypreamble \xeCJKsetwidth
4602 \cs_generate_variant:Nn \tl_map_inline:nn { e }

```

`\xeCJKsetkern` 手动设置相邻标点的距离。

```

4603 \NewDocumentCommand \xeCJKsetkern { m m m }
4604 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
4605 \onlypreamble \xeCJKsetkern

\c__xeCJK_left_tl 4606 \tl_const:Nn \c__xeCJK_left_tl { left }
\c__xeCJK_right_tl 4607 \tl_const:Nn \c__xeCJK_right_tl { right }

```

相关选项声明。

```

AllowBreakBetweenPuncts
KaiMingPunct
LongPunct
NoBreakLongPunct
MiddlePunct
PoZheHaoLigature
LatinPunct
CJLineBreak
PunctWidth
PunctBoundWidth
RubberPunctSkip

4608 \keys_define:nn { xeCJK / options }
4609 {
4610   AllowBreakBetweenPuncts .choice: ,
4611   AllowBreakBetweenPuncts / true .code:n =
4612   {
4613     \bool_set_true:N \l__xeCJK_punct_breakable_bool
4614     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
4615     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
4616     \__xeCJK_punct_bound_breakable_kern:N
4617   } ,
4618   AllowBreakBetweenPuncts / false .code:n =
4619   {
4620     \bool_set_false:N \l__xeCJK_punct_breakable_bool
4621     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_kern:NN
4622     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
4623     \__xeCJK_nobreak_hskip:N
4624   } ,
4625   AllowBreakBetweenPuncts .default:n = { true } ,
4626   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
4627   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
4628   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
4629   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
4630   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
4631   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
4632   NoBreakLongPunct .code:n = { \__xeCJK_set_special_punct:nn { nobreak_long } {#1} } ,
4633   NoBreakLongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { nobreak_long } {#1} } ,
4634   NoBreakLongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { nobreak_long } {#1} } ,
4635   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
4636   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
4637   MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
4638   PoZheHaoLigature .choice: ,
4639   PoZheHaoLigature / true .code:n =
4640   {
4641     \bool_set_true:N \l__xeCJK_pozhehao_ligature_bool
4642     \xeCJK_declare_char_class:nn { PoZheHao } { "2014 , "2015 }
4643   } ,
4644   PoZheHaoLigature / false .code:n =
4645   {
4646     \bool_set_false:N \l__xeCJK_pozhehao_ligature_bool
4647     \xeCJK_declare_char_class:nn { FullRight } { "2014 }
4648     \xeCJK_declare_char_class:nn { Default } { "2015 }
4649   } ,
4650   PoZheHaoLigature .default:n = { true } ,
4651   LatinPunct .choice: ,
4652   LatinPunct / true .code:n =
4653   {
4654     \bool_set_true:N \l__xeCJK_latin_punct_bool
4655     \xeCJK_declare_char_class:nn { HalfLeft }
4656     { "2018 , "201C }
4657     \xeCJK_declare_char_class:nn { HalfRight }
4658     { "00B7 , "2019 , "201D , "2025 , "2026 , "2027 }
4659   } ,
4660   LatinPunct / false .code:n =
4661   {
4662     \bool_set_false:N \l__xeCJK_latin_punct_bool
4663     \xeCJK_declare_char_class:nn { FullLeft }
4664     { "2018 , "201C }
4665     \xeCJK_declare_char_class:nn { FullRight }
4666     { "00B7 , "2019 , "201D , "2025 , "2026 , "2027 }
4667   } ,
4668   LatinPunct .default:n = { true } ,
4669   CJLineBreak .choices:nn = { normal , strict }
4670   {
4671     \str_if_eq:nnTF {#1} { strict }
4672     {
4673       \bool_set_true:N \l__xeCJK_CJ_strict_bool

```

```

4674         \xeCJK_declare_char_class:nN { CJStarter } \c__xeCJK_CJ_chars_clist
4675     }
4676     {
4677         \bool_set_false:N \l__xeCJK_CJ_strict_bool
4678         \xeCJK_declare_char_class:nN { CJK } \c__xeCJK_CJ_chars_clist
4679     }
4680 },
4681 CJLineBreak .initial:n = { normal } ,
4682 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
4683 PunctBoundWidth .tl_gset:N = \g__xeCJK_punct_bound_width_tl ,
4684 PunctWidth .value_required:n = true ,
4685 PunctBoundWidth .value_required:n = true ,
4686 RubberPunctSkip .choice: ,
4687 RubberPunctSkip .default:n = { true } ,
4688 RubberPunctSkip / true .code:n =
4689     { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN } ,
4690 RubberPunctSkip / plus .code:n =
4691     { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_plus:nNN } ,
4692 RubberPunctSkip / minus .code:n =
4693     { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_minus:nNN } ,
4694 RubberPunctSkip / false .code:n =
4695     { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_dim:nNN }
4696 }
4697 \bool_new:N \l__xeCJK_punct_breakable_bool

```

相关选项定义的辅助函数。

```

4698 \clist_new:N \g__xeCJK_special_punct_clist
4699 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , nobreak_long , middle }
4700 \cs_new:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
4701 \cs_new:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
4702 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
4703     { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
4704 \cs_new_protected:Npn \__xeCJK_set_special_punct:nn #1#2
4705     {
4706         \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
4707             { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
4708         \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
4709         \tl_map_inline:en {#2}
4710             {
4711                 \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4712                 \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4713             }
4714     }
4715 \cs_new_protected:Npn \__xeCJK_add_special_punct:nn #1#2
4716     {
4717         \tl_map_inline:en {#2}
4718             {
4719                 \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
4720                 {
4721                     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4722                     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4723                 }
4724             }
4725     }
4726 \cs_new_protected:Npn \__xeCJK_sub_special_punct:nn #1#2
4727     {
4728         \tl_map_inline:en {#2}
4729             {
4730                 \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4731                 \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4732             }
4733     }

```

判断一个标点符号是否为全角右标点和长标点符号。注意, PoZheHao 类的破折号同样按全角右标点处理, 保证它与其他标点相邻时使用正确的间距参数。

```

4734 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }

```

```

4735 {
4736   \if_int_compare:w \xeCJK_token_value_class:N #1 =
4737     \xeCJK_class_num:n { FullRight }
4738     \prg_return_true:
4739   \else:
4740     \if_int_compare:w \xeCJK_token_value_class:N #1 =
4741       \xeCJK_class_num:n { PoZheHao }
4742       \prg_return_true:
4743     \else:
4744       \prg_return_false:
4745     \fi:
4746   \fi:
4747 }
4748 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
4749 {
4750   \exp_args:Nc
4751   \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
4752   {
4753     \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
4754     \prg_return_true: \else: \prg_return_false: \fi:
4755   }
4756 }

```

一些用于记录的辅助函数。

```

4757 \cs_new:Npn \__xeCJK_punct_csname:n #1
4758 { c__xeCJK_\l_xeCJK_current_punct_font_tl/\l_xeCJK_punct_style_tl/#1/tl }
4759 \cs_new:Npn \__xeCJK_use_punct_dim:nN #1#2
4760 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2 } } }
4761 \cs_new:Npn \__xeCJK_use_punct_dim:nNN #1#2#3
4762 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2/#3 } } }
4763 \cs_new:Npn \__xeCJK_use_punct_skip:nNN #1#2#3
4764 { \use:c { \__xeCJK_punct_csname:n { skip/#1/#2/#3 } } }
4765 \cs_new:Npn \__xeCJK_use_punct_skip_plus:nNN #1#2#3
4766 { \use:c { \__xeCJK_punct_csname:n { skip/plus/#1/#2/#3 } } }
4767 \cs_new:Npn \__xeCJK_use_punct_skip_minus:nNN #1#2#3
4768 { \use:c { \__xeCJK_punct_csname:n { skip/minus/#1/#2/#3 } } }
4769 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNn #1#2
4770 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2 } }
4771 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNNn #1#2#3
4772 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2/#3 } }
4773 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNn #1#2#3#4
4774 {
4775   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { #1/#2/#3 } {#4}
4776   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { plus/#1/#2/#3 } {#4}
4777   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { minus/#1/#2/#3 } {#4}
4778 }
4779 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNnnn #1#2#3#4#5#6
4780 {
4781   \exp_last_unbraced:Ne
4782   \__xeCJK_save_punct_skip_aux:nnnnn
4783   {
4784     {#1}
4785     { #1/#2/#3 }
4786     { \dim_eval:n {#4} }
4787     { \dim_max:nn { \c_zero_dim } {#5} }
4788     { \dim_max:nn { \c_zero_dim } {#6} }
4789   }
4790 }
4791 \cs_new_protected:Npn \__xeCJK_save_punct_skip_aux:nnnnn #1#2#3#4#5
4792 {
4793   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4794   {#2} { #3 ~ plus ~ #4 ~ minus ~ #5 ~ }
4795   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4796   { plus/#2 } { #3 ~ plus ~ #4 ~ }
4797   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4798   { minus/#2 } { #3 ~ minus ~ #5 ~ }
4799 }

```

```

4800 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:nnnn #1#2#3#4
4801 {
4802   \__xeCJK_save_punct_width_aux:cen
4803   { \__xeCJK_punct_csname:n { #1/#3 } }
4804   { \use:c { #1_eval:n } {#4} }
4805   {#2}
4806 }
4807 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:Nnn #1#2#3
4808 {
4809   \tl_const:Nn #1 {#2}
4810   \str_if_eq:nnT {#3} { glue }
4811   { \prop_gput:Nnn \g__xeCJK_punct_skip_prop {#2} { } }
4812 }
4813 \prop_new:N \g__xeCJK_punct_skip_prop
4814 \prop_gput:Non \g__xeCJK_punct_skip_prop { \skip_use:N \c_zero_skip } { }
4815 \cs_generate_variant:Nn \__xeCJK_save_punct_width_aux:Nnn { ce }
4816 \cs_new_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN

```

定义标点处理模板。

```

4817 \DeclareObjectType { xeCJK / punctuation } { 0 }
4818 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { 0 }
4819 {
4820   enabled-global-setting      : boolean = true ,
4821   fixed-punct-width          : length  = \c_max_dim ,
4822   fixed-punct-ratio          : real    = \c_one_fp ,
4823   mixed-punct-width          : length  = \KeyValue { fixed-punct-width } ,
4824   mixed-punct-ratio          : real    = \KeyValue { fixed-punct-ratio } ,
4825   middle-punct-width         : length  = \KeyValue { fixed-punct-width } ,
4826   middle-punct-ratio         : real    = \KeyValue { fixed-punct-ratio } ,
4827   fixed-margin-width         : length  = \c_max_dim ,
4828   fixed-margin-ratio         : real    = \c_one_fp ,
4829   mixed-margin-width         : length  = \KeyValue { fixed-margin-width } ,
4830   mixed-margin-ratio         : real    = \KeyValue { fixed-margin-ratio } ,
4831   middle-margin-width        : length  = \KeyValue { fixed-margin-width } ,
4832   middle-margin-ratio        : real    = \KeyValue { fixed-margin-ratio } ,
4833   bound-punct-width          : length  = \c_max_dim ,
4834   bound-punct-ratio          : real    = \c_nan_fp ,
4835   bound-margin-width         : length  = \c_max_dim ,
4836   bound-margin-ratio         : real    = \c_zero_fp ,
4837   enabled-hanging            : boolean = false ,
4838   add-min-bound-to-margin    : boolean = false ,
4839   optimize-margin            : boolean = false ,
4840   margin-minimum             : length  = \c_zero_dim ,
4841   enabled-kerning            : boolean = true ,
4842   enabled-left-right-kerning : boolean = true ,
4843   min-bound-to-kerning       : boolean = false ,
4844   kerning-total-width        : length  = \c_max_dim ,
4845   kerning-total-ratio        : real    = 0.75 ,
4846   optimize-kerning           : boolean = false ,
4847   same-align-margin          : length  = \c_max_dim ,
4848   same-align-ratio           : real    = \c_nan_fp ,
4849   different-align-margin     : length  = \c_max_dim ,
4850   different-align-ratio      : real    = \c_nan_fp ,
4851   kerning-margin-width       : length  = \c_max_dim ,
4852   kerning-margin-ratio       : real    = \c_one_fp ,
4853   kerning-margin-minimum     : length  = \c_zero_dim
4854 }
4855 \DeclareTemplateCode { xeCJK / punctuation } { basic } { 0 }
4856 {
4857   enabled-global-setting      = \l__xeCJK_enabled_global_setting_bool ,
4858   fixed-punct-width          = \l__xeCJK_fixed_punct_width_dim ,
4859   fixed-punct-ratio          = \l__xeCJK_fixed_punct_ratio_fp ,
4860   mixed-punct-width          = \l__xeCJK_mixed_punct_width_dim ,
4861   mixed-punct-ratio          = \l__xeCJK_mixed_punct_ratio_fp ,
4862   middle-punct-width         = \l__xeCJK_middle_punct_width_dim ,
4863   middle-punct-ratio         = \l__xeCJK_middle_punct_ratio_fp ,

```

```

4864 fixed-margin-width = \l__xeCJK_fixed_margin_width_dim ,
4865 fixed-margin-ratio = \l__xeCJK_fixed_margin_ratio_fp ,
4866 mixed-margin-width = \l__xeCJK_mixed_margin_width_dim ,
4867 mixed-margin-ratio = \l__xeCJK_mixed_margin_ratio_fp ,
4868 middle-margin-width = \l__xeCJK_middle_margin_width_dim ,
4869 middle-margin-ratio = \l__xeCJK_middle_margin_ratio_fp ,
4870 bound-punct-width = \l__xeCJK_bound_punct_width_dim ,
4871 bound-punct-ratio = \l__xeCJK_bound_punct_ratio_fp ,
4872 bound-margin-width = \l__xeCJK_bound_margin_width_dim ,
4873 bound-margin-ratio = \l__xeCJK_bound_margin_ratio_fp ,
4874 enabled-hanging = \l__xeCJK_enabled_hanging_bool ,
4875 add-min-bound-to-margin = \l__xeCJK_add_min_bound_to_margin_bool ,
4876 optimize-margin = \l__xeCJK_optimize_margin_bool ,
4877 margin-minimum = \l__xeCJK_margin_minimum_dim ,
4878 enabled-kerning = \l__xeCJK_enabled_kerning_bool ,
4879 enabled-left-right-kerning = \l__xeCJK_enabled_left_right_kerning_bool ,
4880 min-bound-to-kerning = \l__xeCJK_min_bound_to_kerning_bool ,
4881 kerning-total-width = \l__xeCJK_kerning_total_width_dim ,
4882 kerning-total-ratio = \l__xeCJK_kerning_total_ratio_fp ,
4883 optimize-kerning = \l__xeCJK_optimize_kerning_bool ,
4884 same-align-margin = \l__xeCJK_same_align_margin_dim ,
4885 same-align-ratio = \l__xeCJK_same_align_ratio_fp ,
4886 different-align-margin = \l__xeCJK_different_align_margin_dim ,
4887 different-align-ratio = \l__xeCJK_different_align_ratio_fp ,
4888 kerning-margin-width = \l__xeCJK_kerning_margin_width_dim ,
4889 kerning-margin-ratio = \l__xeCJK_kerning_margin_ratio_fp ,
4890 kerning-margin-minimum = \l__xeCJK_kerning_margin_minimum_dim
4891 }
4892 { \AssignTemplateKeys }

```

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

```

4893 \cs_new_protected:Npn \xeCJK_get_punct_bounds:NN #1#2
4894 {
4895   \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/glue/#1/#2 } }
4896   { \__xeCJK_get_punct_bounds_aux:NN #1 #2 }
4897 }
4898 \cs_new_protected:Npn \__xeCJK_get_punct_bounds_aux:NN
4899 {
4900   \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
4901   { \__xeCJK_save_punct_margin_plain:NN }
4902   { \__xeCJK_save_punct_margin:NN }
4903 }
4904 \cs_new_protected:Npn \xeCJK_get_punct_bounds:No
4905 { \exp_last_unbraced:NNo \xeCJK_get_punct_bounds:NN }
4906 \cs_new_protected:Npn \__xeCJK_save_punct_margin_plain:NN #1#2
4907 {
4908   \__xeCJK_save_punct_dim:nNNn { glue } #1 #2 { \c_zero_dim }
4909   \__xeCJK_save_punct_dim:nNNn { offset } #1 #2 { \c_zero_dim }
4910   \__xeCJK_save_punct_dim:nNNn { margin } #1 #2 { \c_zero_dim }
4911   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_left_tl {#2} { \c_zero_dim }
4912   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_right_tl {#2} { \c_zero_dim }
4913   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
4914   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
4915   \__xeCJK_save_punct_skip:nNNn { glue } #1 #2 { \c_zero_skip }
4916 }
4917 \cs_new_protected:Npn \__xeCJK_save_punct_margin:NN #1#2
4918 {
4919   \group_begin:
4920     \xeCJK_select_punct_font:
4921     \xeCJK_fallback_punct_symbol:NN
4922     \xeCJK_calc_punct_dimen:N #2
4923   \group_end:
4924   \dim_set:Nn \l__xeCJK_bound_dim
4925   { \__xeCJK_use_punct_dim:nNN { bound } #1 #2 }
4926   \tl_if_eq:NNTF #1 \c__xeCJK_right_tl
4927   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_left_tl }
4928   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_right_tl }

```

```

4929 \dim_set:Nn \l__xeCJK_reverse_bound_dim
4930 { \__xeCJK_use_punct_dim:nNN { bound } \l__xeCJK_reverse_tl #2 }
4931 \UseInstance { xeCJK / punctuation } { \l__xeCJK_punct_style_tl }
4932 \xeCJK_punct_margin_process:NN #1 #2
4933 \xeCJK_punct_offset_process:NN #1 #2
4934 \__xeCJK_punct_if_long:NT #2
4935 { \__xeCJK_long_punct_kerning:N #2 }
4936 }
4937 \tl_new:N \l__xeCJK_reverse_tl
4938 \dim_new:N \l__xeCJK_bound_dim
4939 \dim_new:N \l__xeCJK_reverse_bound_dim

```

__xeCJK_long_punct_kerning:N 相同长标点压缩。对于破折号(U+2014)，计算两标点之间的空白，保证它中间不被断开，同时保证其总宽(字面或字框)不超过两个汉字宽。压缩量取以下三者的最大值(width为字框宽，dimen为字面宽，F为当前字号)：

- bound之和，用于字面窄于字框的字体(如中易宋体)；
- $\text{dimen} + \text{width} - 2F$ ，用于字面溢出字框的字体(如方正兰亭黑)；
- $2\text{width} - 2F$ ，用于字框宽于字号的字体(如微软雅黑)。

注意，破折号的边界可能为负值(比如方正新书宋)，此时不必压缩，故取值不小于零。省略号(U+2025、U+2026)连用时不需要压缩，其余长标点保持按边空压缩。

```

4940 \cs_new_protected:Npn \__xeCJK_long_punct_kerning:N #1
4941 {
4942   \dim_set:Nn \l__xeCJK_tmp_dim
4943   {
4944     \dim_max:nn
4945     { \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim }
4946     { \c_zero_dim }
4947   }
4948   \__xeCJK_save_punct_dim:nNNn { bound_width } #1 #1 { \l__xeCJK_tmp_dim }
4949   \dim_set:Nn \l__xeCJK_tmp_dim
4950   {
4951     \str_case:nnF {#1}
4952     {
4953       { ~~~~2014 }
4954       {
4955         - \dim_max:nn
4956         { \l__xeCJK_tmp_dim }
4957         {
4958           \dim_max:nn
4959           {
4960             \__xeCJK_use_punct_dim:nN { dimen } #1 +
4961             \__xeCJK_use_punct_dim:nN { width } #1 -
4962             \f@size pt - \f@size pt
4963           }
4964           {
4965             \__xeCJK_use_punct_dim:nN { width } #1 +
4966             \__xeCJK_use_punct_dim:nN { width } #1 -
4967             \f@size pt - \f@size pt
4968           }
4969         }
4970       }
4971       { ~~~~2025 } { \c_zero_dim }
4972       { ~~~~2026 } { \c_zero_dim }
4973     }
4974     { -\l__xeCJK_tmp_dim }
4975   }
4976   \__xeCJK_save_punct_dim:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
4977   \__xeCJK_save_punct_skip:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
4978   \dim_add:Nn \l__xeCJK_tmp_dim
4979   { \dim_max:nn { \l__xeCJK_bound_dim } { \c_zero_dim } }
4980   \__xeCJK_save_punct_dim:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
4981   \__xeCJK_save_punct_skip:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
4982 }

```

\xeCJK_get_punct_kerning:NN 标点压缩。若当前格式关闭左标点到右标点的压缩,则保存零宽且无伸缩量的间距;用户通过
 _xeCJK_save_punct_kerning_plain:NN \xeCJKsetkern 显式设置的字符对仍然优先,除非当前格式同时关闭全局设置。

```

4983 \cs_new_protected:Npn \xeCJK_get_punct_kerning:NN #1#2
4984 {
4985   \tl_if_exist:cF { \_xeCJK_punct_csname:n { dim/kern/#1/#2 } }
4986   {
4987     \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c_xeCJK_punct_style_plain_tl
4988     { \_xeCJK_save_punct_kerning_plain:NN }
4989     { \_xeCJK_save_punct_kerning:NN }
4990     #1 #2
4991   }
4992 }
4993 \cs_new_protected:Npn \xeCJK_get_punct_kerning:oN
4994 { \exp_after:wN \xeCJK_get_punct_kerning:NN }
4995 \cs_new_protected:Npn \_xeCJK_save_punct_kerning_plain:NN #1#2
4996 {
4997   \_xeCJK_save_punct_dim:nNNn { kern } #1 #2 { \c_zero_dim }
4998   \_xeCJK_save_punct_dim:nNNn { bound_kern } #1 #2 { \c_zero_dim }
4999   \_xeCJK_save_punct_dim:nNNn { bound_width } #1 #2 { \c_zero_dim }
5000   \_xeCJK_save_punct_skip:nNNn { kern } #1 #2 { \c_zero_skip }
5001   \_xeCJK_save_punct_skip:nNNn { bound_kern } #1 #2 { \c_zero_skip }
5002 }
5003 \cs_new_protected:Npn \_xeCJK_save_punct_kerning:NN #1#2
5004 {
5005   % 后续分支需要读取当前实例的布尔选项,因此必须在分支前载入实例。
5006   \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
5007   \_xeCJK_punct_if_right:NTF #1
5008   { \xeCJK_punct_kerning_process:NN #1 #2 }
5009   {
5010     \_xeCJK_punct_if_right:NTF #2
5011     { \_xeCJK_save_left_right_kerning:NN #1 #2 }
5012     { \xeCJK_punct_kerning_process:NN #1 #2 }
5013   }
5014 }
5015 \cs_new_protected:Npn \_xeCJK_save_left_right_kerning:NN #1#2
5016 {
5017   \bool_if:NTF \l_xeCJK_enabled_left_right_kerning_bool
5018   { \xeCJK_punct_kerning_process:NN #1 #2 }
5019   {
5020     \bool_if:NTF \l_xeCJK_enabled_global_setting_bool
5021     {
5022       \tl_if_exist:cTF { g_xeCJK_punct/kern/#1/#2/tl }
5023       { \xeCJK_punct_kerning_process:NN #1 #2 }
5024       { \_xeCJK_save_punct_kerning_plain:NN #1 #2 }
5025     }
5026     { \_xeCJK_save_punct_kerning_plain:NN #1 #2 }
5027   }
5028 }

```

_xeCJK_punct_if_full_margin_dash:NN 判断标点 #1 是否为需要全空白补偿的破折号,即 U+2014 且未被归入 PoZheHao 类(没有启用破折号合字)。这类破折号在连用时,两个 U+2014 之间被挤压掉的空白恰好等于单个字符左右的总空白,因此在其两端各补偿一整份空白(而非一半),可以优先保证破折号占两个汉字宽。

```

5029 \prg_new_conditional:Npnn \_xeCJK_punct_if_full_margin_dash:N #1 { p , T , F , TF }
5030 {
5031   \str_if_eq:nnTF {#1} { ^^^2014 }
5032   {
5033     \if_int_compare:w \xeCJK_token_value_class:N #1 =
5034     \xeCJK_class_num:n { PoZheHao }
5035     \prg_return_false:
5036   }
5037   \else:
5038     \prg_return_true:
5039   \fi:
5040   { \prg_return_false: }
5041 }

```

\xeCJK_punct_margin_process:NN 注意, 未启用合字的破折号(U+2014)两端各补偿一整份空白。此时单独一个 U+2014 占一个字宽加一份空白, 连用的两个 U+2014 恰好占两个字宽。

```

5042 \cs_new_protected:Npn \xeCJK_punct_margin_process:NN #1#2
5043 {
5044   \dim_set:Nn \l__xeCJK_tmp_dim
5045   {
5046     \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5047     {
5048       \cs_if_exist_use:cF { g__xeCJK_punct_width/#2/tl }
5049       {
5050         \tl_if_empty:NTF \g__xeCJK_punct_width_tl
5051         { \__xeCJK_calc_punct_width:N #2 }
5052         { \g__xeCJK_punct_width_tl }
5053       }
5054     }
5055     { \__xeCJK_calc_punct_width:N #2 }
5056   }
5057   \dim_set:Nn \l__xeCJK_margin_dim
5058   {
5059     \dim_max:nn
5060     { \l__xeCJK_margin_minimum_dim }
5061     {
5062       \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
5063       {
5064         \__xeCJK_punct_if_middle:NTF #2
5065         {
5066           ( \l__xeCJK_tmp_dim
5067             - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5068           )
5069           \__xeCJK_punct_if_full_margin_dash:NF #2 { / 2 }
5070         }
5071         {
5072           \bool_if:NTF \l__xeCJK_optimize_margin_bool
5073           {
5074             \dim_max:nn
5075             {
5076               \dim_min:nn
5077               { \l__xeCJK_bound_dim }
5078               { \l__xeCJK_reverse_bound_dim }
5079             }
5080             { \use:n }
5081             {
5082               \l__xeCJK_tmp_dim
5083               - \l__xeCJK_reverse_bound_dim
5084               - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5085             }
5086           }
5087         }
5088       }
5089       {
5090         \bool_if:NTF \l__xeCJK_optimize_margin_bool
5091         { \dim_min:nn { \l__xeCJK_bound_dim } }
5092         { \use:n }
5093         { \__xeCJK_calc_margin_width:N #2 }
5094       }
5095     }
5096   }
5097   \__xeCJK_save_punct_dim:nNn { margin } #1 #2 { \l__xeCJK_margin_dim }
5098 }
5099 \dim_new:N \l__xeCJK_margin_dim

```

__xeCJK_calc_punct_width:N 5100 \cs_new:Npn __xeCJK_calc_punct_width:N #1

```

5101 {
5102   \__xeCJK_punct_if_middle:NTF #1
5103   { \__xeCJK_punct_width_or_ratio:nN { middle } }
5104   {
5105     \__xeCJK_punct_if_mixed_width:NTF #1

```

```

5106         { \_xeCJK_punct_width_or_ratio:nN { mixed } }
5107         { \_xeCJK_punct_width_or_ratio:nN { fixed } }
5108     }
5109     #1
5110 }

\_xeCJK_calc_margin_width:N 5111 \cs_new:Npn \_xeCJK_calc_margin_width:N #1
5112 {
5113     \_xeCJK_punct_if_middle:NTF #1
5114     {
5115         \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
5116         { \l__xeCJK_middle_margin_width_dim }
5117         {
5118             \_xeCJK_dim_ratio:Nn \l__xeCJK_middle_margin_ratio_fp
5119             { ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / 2 }
5120         }
5121     }
5122     {
5123         \_xeCJK_punct_if_mixed_width:NTF #1
5124         { \_xeCJK_margin_width_or_ratio:n { mixed } }
5125         { \_xeCJK_margin_width_or_ratio:n { fixed } }
5126     }
5127 }

\_xeCJK_dim_ratio:Nn 5128 \cs_new:Npn \_xeCJK_dim_ratio:Nn #1#2
5129 { \fp_to_dim:n { #1 \dim_to_fp:n {#2} } }
5130 \cs_generate_variant:Nn \_xeCJK_dim_ratio:Nn { c }

\_xeCJK_punct_offset_process:NN 5131 \cs_new_protected:Npn \xeCJK_punct_offset_process:NN #1#2
5132 {
5133     \dim_set:Nn \l__xeCJK_tmp_dim
5134     {
5135         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5136         {
5137             \cs_if_exist_use:cF { g__xeCJK_punct_bound_width/#2/tl }
5138             {
5139                 \tl_if_empty:NTF \g__xeCJK_punct_bound_width_tl
5140                 { \_xeCJK_punct_width_or_ratio:nN { bound } #2 }
5141                 { \g__xeCJK_punct_bound_width_tl }
5142             }
5143         }
5144         { \_xeCJK_punct_width_or_ratio:nN { bound } #2 }
5145     }
5146     \dim_set:Nn \l__xeCJK_tmp_dim
5147     {
5148         \bool_if:NTF \l__xeCJK_enabled_hanging_bool
5149         { \use:n }
5150         { \dim_max:nn { \l__xeCJK_margin_minimum_dim } }
5151         {
5152             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
5153             {
5154                 \_xeCJK_punct_if_middle:NTF #2
5155                 {
5156                     \l__xeCJK_tmp_dim
5157                     - \l__xeCJK_margin_dim
5158                     - ( \_xeCJK_use_punct_dim:nN { dimen } #2 )
5159                 }
5160                 {
5161                     \l__xeCJK_tmp_dim
5162                     - \l__xeCJK_reverse_bound_dim
5163                     - ( \_xeCJK_use_punct_dim:nN { dimen } #2 )
5164                 }
5165             }
5166         }
5167         \bool_if:NTF \l__xeCJK_optimize_margin_bool
5168         { \dim_min:nn { \l__xeCJK_bound_dim } }
5169         { \use:n }

```

```

5170         { \__xeCJK_margin_width_or_ratio:n { bound } }
5171     }
5172 }
5173 }
5174 \__xeCJK_save_punct_dim:nNNn { offset } #1 #2
5175 { \l__xeCJK_tmp_dim }
5176 \__xeCJK_save_punct_dim:nNNn { rule } #1 #2
5177 { \l__xeCJK_tmp_dim - \l__xeCJK_bound_dim }
5178 \__xeCJK_save_punct_dim:nNNn { rule } \l__xeCJK_reverse_tl #2
5179 { \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim }
5180 \__xeCJK_save_punct_dim:nNNn { glue } #1 #2
5181 { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
5182 \__xeCJK_save_punct_skip:nNNnn { glue } #1 #2
5183 { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
5184 {
5185     \__xeCJK_punct_if_middle:NTF #2
5186     {
5187         ( \__xeCJK_use_punct_dim:n { width } #2 -
5188           \__xeCJK_use_punct_dim:n { dimen } #2 )
5189         \__xeCJK_punct_if_full_margin_dash:NF #2 { / 2 }
5190         - \l__xeCJK_margin_dim
5191     }
5192     { \l__xeCJK_bound_dim - \l__xeCJK_margin_dim }
5193 }
5194 {
5195     \__xeCJK_punct_if_middle:NTF #2
5196     { \l__xeCJK_margin_dim / 2 }
5197     { \l__xeCJK_margin_dim - \l__xeCJK_reverse_bound_dim }
5198 }
5199 }

```

```

\__xeCJK_punct_width_or_ratio:n 5200 \cs_new:Npn \__xeCJK_punct_width_or_ratio:n #1#2
5201 {
5202     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
5203     { \use:c { l__xeCJK_#1_punct_width_dim } }
5204     {
5205         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_punct_ratio_fp } }
5206         { \c_max_dim }
5207         {
5208             \__xeCJK_dim_ratio:cn
5209             { l__xeCJK_#1_punct_ratio_fp }
5210             { \__xeCJK_use_punct_dim:n { width } #2 }
5211         }
5212     }
5213 }

```

```

\__xeCJK_margin_width_or_ratio:n 5214 \cs_new:Npn \__xeCJK_margin_width_or_ratio:n #1
5215 {
5216     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
5217     { \use:c { l__xeCJK_#1_margin_width_dim } }
5218     {
5219         \__xeCJK_dim_ratio:cn
5220         { l__xeCJK_#1_margin_ratio_fp }
5221         { \l__xeCJK_bound_dim }
5222     }
5223     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
5224     { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
5225 }

```

\xeCJK_punct_kerning_process:NN 当标点之一为长标点时, 不必进行压缩。

```

5226 \cs_new_protected:Npn \xeCJK_punct_kerning_process:NN #1#2
5227 {
5228     \dim_set:Nn \l__xeCJK_margin_dim
5229     { \__xeCJK_original_kerning_margin:NN #1 #2 }
5230     \dim_set:Nn \l__xeCJK_minimum_bound_dim
5231     { \__xeCJK_punct_min_bound:NN #1 #2 }

```

```

5232 \__xeCJK_punct_if_long:NTF #1
5233 { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
5234 {
5235   \__xeCJK_punct_if_long:NT #2
5236   { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
5237 }
5238 \dim_set:Nn \l__xeCJK_kerning_margin_dim
5239 {
5240   \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5241   {
5242     \cs_if_exist_use:cF { g__xeCJK_punct/kern/#1/#2/tl }
5243     { \__xeCJK_punct_kerning_process_aux:NN #1 #2 }
5244   }
5245   { \__xeCJK_punct_kerning_process_aux:NN #1 #2 }
5246 }
5247 \__xeCJK_save_kerning:nnNN { kern } { bound } #1 #2
5248 \__xeCJK_save_punct_dim:nnNn { bound_width } #1 #2
5249 { \l__xeCJK_kerning_margin_dim - \l__xeCJK_tmp_dim }
5250 \__xeCJK_punct_if_right:NTF #1
5251 {
5252   \__xeCJK_punct_if_right:NTF #2
5253   { \__xeCJK_save_kerning:nnnNN { bound_kern } { offset } { bound } }
5254   { \__xeCJK_save_kerning:nnNN { bound_kern } { offset } }
5255 }
5256 {
5257   \__xeCJK_punct_if_right:NTF #2
5258   { \__xeCJK_save_kerning:nnNN { bound_kern } { bound } }
5259   { \__xeCJK_save_kerning:nnnNN { bound_kern } { bound } { offset } }
5260 }
5261 #1 #2
5262 }
5263 \cs_new:Npn \__xeCJK_punct_kerning_process_aux:NN #1#2
5264 {
5265   \bool_if:NTF \l__xeCJK_enabled_kerning_bool
5266   { \__xeCJK_calc_kerning_margin:NN #1 #2 }
5267   { \l__xeCJK_margin_dim }
5268 }
5269 \dim_new:N \l__xeCJK_minimum_bound_dim
5270 \dim_new:N \l__xeCJK_kerning_margin_dim

```

__xeCJK_save_kerning:nnNN 相邻两个标点符号的间距能伸长到原始空白(未压缩时的状态),能收缩到较小边距。

```

5271 \cs_new_protected:Npn \__xeCJK_save_kerning:nnNN #1#2
5272 { \__xeCJK_save_kerning:nnnNN {#1} {#2} {#2} }
5273 \cs_new_protected:Npn \__xeCJK_save_kerning:nnnNN #1#2#3#4#5
5274 {
5275   \dim_set:Nn \l__xeCJK_tmp_dim
5276   {
5277     \l__xeCJK_kerning_margin_dim
5278     - ( \__xeCJK_use_punct_dim:nnN {#2} \c__xeCJK_right_tl #4 )
5279     - ( \__xeCJK_use_punct_dim:nnN {#3} \c__xeCJK_left_tl #5 )
5280   }
5281   \__xeCJK_save_punct_dim:nnNn {#1} #4 #5 { \l__xeCJK_tmp_dim }
5282   \__xeCJK_save_punct_skip:nnNnn {#1} #4 #5
5283   { \l__xeCJK_tmp_dim }
5284   { \l__xeCJK_margin_dim - \l__xeCJK_kerning_margin_dim }
5285   { \l__xeCJK_kerning_margin_dim - \l__xeCJK_minimum_bound_dim }
5286 }

```

__xeCJK_original_kerning_margin:NN 相邻两个标点符号之间的本来空白宽度。

```

5287 \cs_new:Npn \__xeCJK_original_kerning_margin:NN #1#2
5288 {
5289   \dim_eval:n
5290   {
5291     \__xeCJK_use_punct_dim:nnN
5292     { \__xeCJK_punct_if_right:NTF #1 { margin } { bound } } \c__xeCJK_right_tl #1

```

```

5293         +
5294         \__xeCJK_use_punct_dim:nNN
5295         { \__xeCJK_punct_if_right:NTF #2 { bound } { margin } } \c__xeCJK_left_tl #2
5296     }
5297 }

```

```

\__xeCJK_calc_kerning_margin:NN 5298 \cs_new:Npn \__xeCJK_calc_kerning_margin:NN #1#2
\__xeCJK_calc_kerning_margin_aux:NN 5299 {
5300     \dim_max:nn
5301     { \l__xeCJK_kerning_margin_minimum_dim }
5302     {
5303         \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
5304         { \l__xeCJK_minimum_bound_dim }
5305         {
5306             \bool_if:NTF \l__xeCJK_optimize_kerning_bool
5307             { \dim_max:nn { \l__xeCJK_minimum_bound_dim } }
5308             { \use:n }
5309             { \__xeCJK_calc_kerning_margin_aux:NN #1 #2 }
5310         }
5311     }
5312 }
5313 \cs_new:Npn \__xeCJK_calc_kerning_margin_aux:NN #1#2
5314 {
5315     \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
5316     { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
5317     {
5318         \fp_if_nan:NTF { \l__xeCJK_kerning_total_ratio_fp }
5319         {
5320             \xeCJK_if_same_class:NNTF #1 #2
5321             { \__xeCJK_kerning_width_or_ratio:nNN { same } }
5322             { \__xeCJK_kerning_width_or_ratio:nNN { different } }
5323         }
5324         {
5325             \__xeCJK_calc_kerning_margin:nNN
5326             {
5327                 \__xeCJK_dim_ratio:Nn \l__xeCJK_kerning_total_ratio_fp
5328                 {
5329                     \__xeCJK_use_punct_dim:nN { width } #1 +
5330                     \__xeCJK_use_punct_dim:nN { width } #2
5331                 }
5332             }
5333         }
5334     }
5335     #1 #2
5336 }

```

```

\__xeCJK_kerning_width_or_ratio:nNN 5337 \cs_new:Npn \__xeCJK_kerning_width_or_ratio:nNN #1#2#3
5338 {
5339     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
5340     { \use:c { l__xeCJK_#1_align_margin_dim } }
5341     {
5342         \fp_if_nan:NTF { \use:c { l__xeCJK_#1_align_ratio_fp } }
5343         {
5344             \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
5345             { \l__xeCJK_kerning_margin_width_dim \use_none:n }
5346             { \__xeCJK_dim_ratio:Nn \l__xeCJK_kerning_ratio_fp }
5347         }
5348         { \__xeCJK_dim_ratio:cn { l__xeCJK_#1_align_ratio_fp } }
5349         { \l__xeCJK_margin_dim }
5350     }
5351 }

```

```

\__xeCJK_punct_min_bound:NN 5352 \cs_new:Npn \__xeCJK_punct_min_bound:NN #1#2
5353 {
5354     \dim_max:nn
5355     {
5356         \dim_min:nn
5357         { \__xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 }

```

```

5358         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 }
5359     }
5360     {
5361         \dim_min:nn
5362         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #2 }
5363         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #2 }
5364     }
5365 }

```

_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。

```

5366 \cs_new:Npn \_xeCJK_calc_kerning_margin:nNN #1#2#3
5367 {
5368     \dim_eval:n
5369     {
5370         (#1)
5371         - ( \_xeCJK_use_punct_dim:nNN
5372             { \_xeCJK_punct_if_right:NTF #2 { bound } { margin } }
5373             \c__xeCJK_left_tl #2 )
5374         - ( \_xeCJK_use_punct_dim:nNN
5375             { \_xeCJK_punct_if_right:NTF #3 { margin } { bound } }
5376             \c__xeCJK_right_tl #3 )
5377         - ( \_xeCJK_use_punct_dim:nN { dimen } #2 )
5378         - ( \_xeCJK_use_punct_dim:nN { dimen } #3 )
5379     }
5380 }

```

\xeCJK_calc_punct_dimen:N 计算标点的左右实际边距和实际尺寸。

```

5381 \cs_new_protected:Npn \xeCJK_calc_punct_dimen:N #1
5382 {
5383     \_xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl #1
5384     { \xeCJK_glyph_bounds:NN 1 #1 }
5385     \_xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl #1
5386     { \xeCJK_glyph_bounds:NN 3 #1 }
5387     \_xeCJK_save_punct_dim:nNn { width } #1
5388     { \tex_fontcharwd:D \tex_font:D `#1 }
5389     \_xeCJK_save_punct_dim:nNn { dimen } #1
5390     {
5391         ( \_xeCJK_use_punct_dim:nN { width } #1 )
5392         - ( \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 ) -
5393         ( \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 )
5394     }
5395 }

```

\xeCJK_glyph_bounds:NN 用 \XeTeXglyphbounds 取得标点符号的上下左右空白。

```

5396 \cs_new:Npn \xeCJK_glyph_bounds:NN #1#2
5397 { \tex_XeTeXglyphbounds:D #1 ~ \tex_XeTeXcharglyph:D `#2 \exp_stop_f: }

```

```

PunctStyle 5398 \keys_define:nn { xeCJK / options }
5399 { PunctStyle .code:n = \exp_args:Ne \_xeCJK_set_punct_style:n {#1} }
5400 \cs_new_protected:Npn \_xeCJK_set_punct_style:n #1
5401 {
5402     \IfInstanceExistTF { xeCJK / punctuation } {#1}
5403     { \tl_set:Nn \l_xeCJK_punct_style_tl {#1} }
5404     {
5405         \prop_get:NnNF \c__xeCJK_punct_style_alias_prop
5406         {#1} \l_xeCJK_punct_style_tl
5407         { \_xeCJK_error:ne { punct-style-unknown } {#1} }
5408     }
5409 }
5410 \prop_const_from_keyval:Nn \c__xeCJK_punct_style_alias_prop
5411 {
5412     halfwidth      = banjiao ,
5413     fullwidth      = quanjiao ,
5414     mixedwidth     = kaiming ,
5415     marginkerning  = hangmobanjiao ,

```

```

5416     plain          = plain
5417   }
5418   \tl_new:N \l_xeCJK_punct_style_tl
5419   \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
5420   \__xeCJK_msg_new:nn { punct-style-unknown }
5421   {
5422     Punctuation~style~"#1"~is~unknown. \\\
5423     The~available~styles~are~listed~as~follow.\\
5424     "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq
5425     { ~and~ } { ,~ } { ,~and~ }".\\
5426   }

```

`\__xeCJK_trim_spaces:n` `xparse` 处理函数, 先完全展开参数再删除两边空格。

```

5427 \cs_new_protected:Npn \__xeCJK_trim_spaces:n #1
5428 {
5429   \tl_set:Nc \ProcessedArgument
5430   { \exp_args:Nc \tl_trim_spaces:n {#1} }
5431 }

```

`\xeCJKDeclarePunctStyle` 定义新的标点处理风格, 已经存在的同名风格将被覆盖。

```

5432 \NewDocumentCommand \xeCJKDeclarePunctStyle
5433 { > { \__xeCJK_trim_spaces:n } m m }
5434 {
5435   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5436   { \__xeCJK_warning:ne { punct-style-already-defined } {#1} }
5437   { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
5438   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
5439 }
5440 \seq_new:N \g__xeCJK_punct_style_seq
5441 \__xeCJK_msg_new:nn { punct-style-already-defined }
5442 {
5443   Punctuation~style~"#1"~is~already~defined!. \\\
5444   The~existing~style~of~"#1"~will~be~overwritten.\\
5445 }
5446 \@onlypreamble \xeCJKDeclarePunctStyle

```

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

```

5447 \NewDocumentCommand \xeCJKEditPunctStyle
5448 { > { \__xeCJK_trim_spaces:n } m m }
5449 {
5450   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5451   { \EditInstance { xeCJK / punctuation } {#1} {#2} }
5452   { \__xeCJK_error:ne { punct-style-unknown } {#1} }
5453 }
5454 \@onlypreamble \xeCJKEditPunctStyle

```

默认设置即为全角格式。

```

5455 \xeCJKDeclarePunctStyle { quanjiao }
5456 {
5457   optimize-kerning          = true ,
5458   enabled-left-right-kerning = false
5459 }
5460 \xeCJKDeclarePunctStyle { hangmobanjiang } { enabled-kerning = false }
5461 \xeCJKDeclarePunctStyle { banjiao }
5462 {
5463   fixed-punct-ratio = 0.5 ,
5464   optimize-margin   = true ,
5465   kerning-total-ratio = 0.5 ,
5466   optimize-kerning   = true
5467 }
5468 \xeCJKDeclarePunctStyle { kaiming }
5469 {

```

```

5470     fixed-punct-ratio    = 0.5  ,
5471     mixed-punct-ratio    = 1.0  ,
5472     optimize-margin       = true  ,
5473     kerning-total-ratio   = 0.5  ,
5474     optimize-kerning      = true
5475   }

5476 \xeCJKDeclarePunctStyle { CCT }
5477 {
5478     fixed-punct-ratio    = 0.7  ,
5479     optimize-margin       = true  ,
5480     kerning-total-ratio   = 0.6  ,
5481     optimize-kerning      = true
5482 }

```

5.11 后备字体

AutoFallBack 后备字体的宏包选项声明。

```

5483 \keys_define:nn { xeCJK / options }
5484 {
5485     AutoFallBack .choice: ,
5486     AutoFallBack / true .code:n =
5487     {
5488         \cs_set_eq:NN \xeCJK_fallback_symbol:NN
5489             \__xeCJK_fallback_symbol:NN
5490         \cs_set_eq:NN \xeCJK_fallback_punct_symbol:NN
5491             \__xeCJK_fallback_punct_symbol:NN
5492         \cs_set_eq:NN \xeCJK_clear_fallback_font:
5493             \__xeCJK_clear_fallback_font:
5494     } ,
5495     AutoFallBack / false .code:n =
5496     {
5497         \xeCJK_cs_clear:N \xeCJK_fallback_symbol:NN
5498         \xeCJK_cs_clear:N \xeCJK_fallback_punct_symbol:NN
5499         \xeCJK_cs_clear:N \xeCJK_clear_fallback_font:
5500     } ,
5501     AutoFallBack .default:n = { true } ,
5502     fallback .meta:n = { AutoFallBack = true }
5503 }

```

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。

```

\xeCJK_fallback_symbol:NN
\xeCJK_fallback_punct_symbol:NN
5504 \cs_new_protected:Npn \__xeCJK_fallback_symbol:NN #1#2
5505 {
5506     \xeCJK_reset_fallback_font:
5507     \xeCJK_glyph_if_exist:NF #2
5508     { \__xeCJK_fallback_symbol_aux:NN }
5509     #1#2
5510 }
5511 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol:NN #1#2
5512 {
5513     \xeCJK_reset_fallback_font:
5514     \xeCJK_glyph_if_exist:NF #2
5515     { \__xeCJK_fallback_punct_symbol_aux:NN }
5516     #1#2
5517 }
5518 \cs_new_eq:NN \xeCJK_fallback_symbol:NN \prg_do_nothing:
5519 \cs_new_eq:NN \xeCJK_fallback_punct_symbol:NN \prg_do_nothing:
5520 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:NN
5521 {
5522     \__xeCJK_fallback_symbol_aux:nnNN
5523     { \CJK@family }
5524     { \l_xeCJK_family_tl }
5525 }
5526 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol_aux:NN
5527 {

```

```

5528 \__xeCJK_fallback_symbol_aux:nnNN
5529 { \CJK@punctfamily }
5530 { \l_xeCJK_punct_family_tl }
5531 }
5532 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:nnNN
5533 {
5534 \cs_set_protected:Npe \xeCJK_reset_fallback_font:
5535 {
5536 \tex_the:D \tex_font:D
5537 \xeCJK_clear_fallback_font:
5538 }
5539 \exp_args:Nee \__xeCJK_fallback_loop:nnNN
5540 }
5541 \cs_new_protected:Npn \__xeCJK_clear_fallback_font:
5542 { \cs_set_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing: }
5543 \cs_new_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing:
5544 \cs_new_eq:NN \xeCJK_clear_fallback_font: \prg_do_nothing:

```

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

```

5545 \cs_new_protected:Npn \__xeCJK_fallback_loop:nnNN
5546 {
5547 \cs_set_eq:NN \__xeCJK_fallback_loop:TF \use_i:nn
5548 \__xeCJK_fallback_loop:nnnNN { FallBack }
5549 }
5550 \cs_new_protected:Npn \__xeCJK_fallback_loop:nnnNN #1#2#3
5551 {
5552 \xeCJK_select_fallback_font:nnn {#1} {#2} {#3}
5553 \__xeCJK_fallback_loop:TF
5554 { \__xeCJK_fallback_loop_aux:nnnNN }
5555 { \__xeCJK_fallback_missing_glyph:nnnNN }
5556 {#1} {#2} {#3}
5557 }
5558 \cs_new_protected:Npn \__xeCJK_fallback_loop_aux:nnnNN #1#2#3#4#5
5559 {
5560 \xeCJK_glyph_if_exist:NF #5
5561 { \__xeCJK_fallback_loop:nnnNN { #1/FallBack } {#2} {#3} }
5562 #4#5
5563 }
5564 \cs_new_protected:Npn \__xeCJK_fallback_missing_glyph:nnnNN #1#2#3#4#5
5565 {
5566 \__xeCJK_warning:nnee { missing-glyph } {#1} {#2} {#5}
5567 #4#5
5568 }
5569 \cs_new_protected:Npn \xeCJK_select_fallback_font:nnn #1#2
5570 {
5571 \__xeCJK_select_fallback_font:cnnn
5572 { \__xeCJK_font_csname:n { #2/#1 } } {#1} {#2}
5573 }
5574 \cs_new_protected:Npn \__xeCJK_select_fallback_font:Nnnn #1
5575 {
5576 \cs_if_exist:NF #1
5577 { \__xeCJK_fallback_font_initial:NNnnn }
5578 #1 \use_none:nnn
5579 }
5580 \cs_generate_variant:Nn \__xeCJK_select_fallback_font:Nnnn { c }
5581 \cs_new_protected:Npn \__xeCJK_fallback_font_initial:NNnnn #1#2#3#4#5
5582 {
5583 \xeCJK_family_if_exist:nTF { #5/#3 }
5584 { \__xeCJK_font_initial:Nn #1 { #5/#3 } }
5585 { \__xeCJK_fallback_font_initial_auxi:Nnnn #1 {#5} {#3} {#4} }
5586 #1
5587 }
5588 \cs_new_protected:Npn \__xeCJK_fallback_font_initial_auxi:Nnnn #1
5589 {
5590 \exp_args:NNe \__xeCJK_fallback_font_initial_auxii:Nnnnn

```

```

5591     #1 { \CJKfamilydefault }
5592 }
5593 \cs_new_protected:Npn \__xeCJK_fallback_font_initial_auxii:Nnnnn #1#2#3
5594 {
5595     \str_if_eq:nnTF {#2} {#3}
5596     { \__xeCJK_fallback_loop_end:Nnnn }
5597     { \__xeCJK_fallback_font_initial_auxiii:Nnnn }
5598     #1 {#2}
5599 }
5600 \cs_new_protected:Npn \__xeCJK_fallback_font_initial_auxiii:Nnnn #1#2
5601 {
5602     \xeCJK_family_if_exist:nTF {#2}
5603     { \__xeCJK_fallback_font_initial_auxiv:Nnnn }
5604     { \__xeCJK_fallback_loop_end:Nnnn }
5605     #1 {#2}
5606 }
5607 \cs_new_protected:Npn \__xeCJK_fallback_font_initial_auxiv:Nnnn #1#2#3#4
5608 {
5609     \__xeCJK_font_initial:Nn #1 {#2}
5610     \exp_args:Nc \__xeCJK_fallback_font_initial_auxiii:Nnnn
5611     { \__xeCJK_font_csname:n { #4/#3/FallBack } }
5612     { #2/FallBack } { #3/FallBack } {#4}
5613 }
5614 \cs_new_eq:NN \__xeCJK_fallback_loop:TF \use_i:nn
5615 \cs_new_protected:Npn \__xeCJK_fallback_loop_end:Nnnnn #1#2#3#4
5616 { \cs_gset_eq:NN #1 \__xeCJK_fallback_loop_end: }
5617 \cs_new_protected:Npn \__xeCJK_fallback_loop_end:
5618 { \cs_set_eq:NN \__xeCJK_fallback_loop:TF \use_ii:nn }
5619 \__xeCJK_msg_new:nn { missing-glyph }
5620 {
5621     CJKfamily~\__xeCJK_msg_family_map:n {#2}'~(#{1})~
5622     does~not~contain~glyph~`#{3}'~
5623     ( U + \int_to_Hex:n { `#{3} } )~\msg_line_context:.
5624 }
\setCJKfallbackfamilyfont 5625 \NewDocumentCommand \setCJKfallbackfamilyfont { m o m }
5626 {
5627     \__xeCJK_pass_args:nnnn
5628     { \xeCJK_set_family_fallback:nnn {#1} } {#2} {#3}
5629     { }
5630 }
\xCJK_set_family_fallback:nnn 5631 \cs_new_protected:Npn \xeCJK_set_family_fallback:nnn #1#2#3
5632 {
5633     \group_begin:
5634     \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
5635     \prop_get:NoNF \g__xeCJK_family_font_name_prop
5636     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
5637     { \tl_clear:N \l__xeCJK_font_name_tl }
5638     \clist_set:Nn \l__xeCJK_public_options_clist {#2}
5639     \clist_map_function:nN {#3} \__xeCJK_set_family_fallback:n
5640     \group_end:
5641 }
5642 \cs_new_protected:Npn \__xeCJK_set_family_fallback:n #1
5643 {
5644     \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
5645     \__xeCJK_get_sub_features:on \l__xeCJK_fallback_family_tl {#1}
5646     \clist_concat:NNN \l__xeCJK_sub_font_options_clist
5647     \l__xeCJK_public_options_clist
5648     \l__xeCJK_sub_font_options_clist
5649     \exp_args:Nooo \xeCJK_set_family:nnn
5650     \l__xeCJK_fallback_family_tl
5651     \l__xeCJK_sub_font_options_clist
5652     \l__xeCJK_sub_font_name_tl
5653 }
5654 \tl_new:N \l__xeCJK_fallback_family_tl
5655 \clist_new:N \l__xeCJK_public_options_clist

```

5.12 CJK 字体族声明方式

```

5656 \bool_new:N \g__xeCJK_auto_fake_bold_bool
5657 \bool_new:N \g__xeCJK_auto_fake_slant_bool
5658 \fp_new:N \g__xeCJK_embolden_factor_fp
5659 \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。

```

5660 \keys_define:nn { xeCJK / options }
5661 {
5662   AutoFakeBold .choices:nn = { true , false }
5663   { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_bold_bool } ,
5664   AutoFakeBold / unknown .code:n =
5665   {
5666     \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
5667     \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
5668   } ,
5669   AutoFakeBold .default:n = { true } ,
5670   AutoFakeSlant .choices:nn = { true , false }
5671   { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_slant_bool } ,
5672   AutoFakeSlant / unknown .code:n =
5673   {
5674     \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
5675     \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
5676   } ,
5677   AutoFakeSlant .default:n = { true } ,
5678   EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
5679   SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
5680   BoldFont .meta:n = { AutoFakeBold = true } ,
5681   boldfont .meta:n = { AutoFakeBold = true } ,
5682   SlantFont .meta:n = { AutoFakeSlant = true } ,
5683   slantfont .meta:n = { AutoFakeSlant = true }
5684 }

```

\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。

```

5685 \seq_new:N \g__xeCJK_sub_key_seq
5686 \cs_new_protected:Npn \xeCJK_new_sub_key:n #1
5687 {
5688   \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
5689   \keys_define:nn { xeCJK / features }
5690   {
5691     #1 .code:n =
5692     {
5693       \tl_if_blank:nTF {##1}
5694       {
5695         \prop_clear:N \l__xeCJK_sub_key_prop
5696         \tl_set:N \l__xeCJK_sub_family_name_tl
5697         { \l__xeCJK_family_name_tl /#1 }
5698         \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
5699       }
5700       {
5701         \tl_clear:N \l__xeCJK_sub_family_name_tl
5702         \str_if_eq:nnTF {##1} { * }
5703         { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
5704         { \__xeCJK_get_sub_features:nn {#1} {##1} }
5705       }
5706     } ,
5707     #1 .default:n = { }
5708   }
5709 }

```

```

\__xeCJK_get_sub_features:nn 5710 \cs_new_protected:Npn \__xeCJK_get_sub_features:nn #1#2
\__xeCJK_get_sub_features:w 5711 {
5712   \tl_set:N \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
5713   \clist_clear:N \l__xeCJK_sub_font_options_clist

```

```

5714 \exp_after:wN \__xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
5715 \q_mark [ \q_nil ] \q_mark \q_stop
5716 \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
5717 { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
5718 { \tl_replace_all:Nno \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
5719 \prop_put:Nne \l__xeCJK_sub_key_prop {#1}
5720 {
5721 { \exp_not:o \l__xeCJK_sub_font_options_clist }
5722 { \exp_not:o \l__xeCJK_sub_font_name_tl }
5723 }
5724 }
5725 \cs_new_protected:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
5726 {
5727 \quark_if_nil:nTF {#2}
5728 { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
5729 {
5730 \tl_set:Nc \l__xeCJK_sub_font_name_tl
5731 { \xeCJK_tl_remove_outer_braces:n {#3} }
5732 \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
5733 { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
5734 { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
5735 }
5736 }
5737 \tl_new:N \l__xeCJK_sub_family_name_tl
5738 \tl_new:N \l__xeCJK_sub_font_name_tl
5739 \clist_new:N \l__xeCJK_sub_font_options_clist
5740 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { o }
5741 \cs_generate_variant:Nn \tl_replace_all:Nnn { Nno }

```

FallBack 5742 \xeCJK_new_sub_key:n { FallBack }

BoldFont 调用字体的属性声明, 同 **fontspec** 宏包。
ItalicFont

```

5743 \keys_define:nn { xeCJK / features }
5744 {
5745 BoldFont .tl_set:N = \l__xeCJK_font_name_bf_tl ,
5746 ItalicFont .tl_set:N = \l__xeCJK_font_name_it_tl
5747 }

```

```

AutoFakeBold 5748 \keys_define:nn { xeCJK / features }
AutoFakeSlant 5749 {
5750 AutoFakeBold .choice: ,
5751 AutoFakeBold / true .code:n =
5752 {
5753 \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
5754 \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
5755 } ,
5756 AutoFakeBold / false .code:n =
5757 { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
5758 AutoFakeBold / unknown .code:n =
5759 {
5760 \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
5761 \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
5762 } ,
5763 AutoFakeBold .default:n = { true } ,
5764 AutoFakeSlant .choice: ,
5765 AutoFakeSlant / true .code:n =
5766 {
5767 \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
5768 \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
5769 } ,
5770 AutoFakeSlant / false .code:n =
5771 { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
5772 AutoFakeSlant / unknown .code:n =
5773 {
5774 \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
5775 \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }

```

```

5776     } ,
5777     AutoFakeSlant .default:n = { true }
5778 }

\__xeCJK_set_family_initial: 5779 \cs_new_protected:Npn \__xeCJK_set_family_initial:
5780 {
5781     \int_gincr:N \g__xeCJK_family_int
5782     \prop_clear:N \l__xeCJK_sub_key_prop
5783     \tl_clear:N \l__xeCJK_font_name_bf_tl
5784     \tl_clear:N \l__xeCJK_font_name_it_tl
5785     \tl_clear:N \l__xeCJK_sub_family_name_tl
5786     \clist_clear:N \l__xeCJK_fontspec_options_clist
5787     \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
5788     \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
5789     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
5790     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
5791 }
5792 \int_new:N \g__xeCJK_family_int
5793 \prop_new:N \l__xeCJK_sub_key_prop
5794 \clist_new:N \l__xeCJK_fontspec_options_clist
5795 \bool_new:N \l__xeCJK_auto_fake_bold_bool
5796 \bool_new:N \l__xeCJK_auto_fake_slant_bool
5797 \fp_new:N \l__xeCJK_embolden_factor_fp
5798 \fp_new:N \l__xeCJK_slant_factor_fp

```

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

```

5799 \cs_new_protected:Npn \xeCJK_set_family:nnn #1#2#3
5800 {
5801     \group_begin:
5802     \__xeCJK_set_family_initial:
5803     \tl_set:Nn \l__xeCJK_family_name_tl {#1}
5804     \clist_set:Nn \l__xeCJK_font_options_clist {#2}
5805     \tl_set:Nn \l__xeCJK_font_name_tl {#3}
5806     \clist_concat:NNN \l__xeCJK_font_options_clist
5807     \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
5808     \keys_set_known:noN { xeCJK / features }
5809     \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
5810     \__xeCJK_binding_sub_family:
5811     \__xeCJK_parse_font_shape:
5812     \__xeCJK_check_family:o \l__xeCJK_family_name_tl
5813     \__xeCJK_gset_family_cs:n { \l__xeCJK_family_name_tl }
5814     \__xeCJK_save_family_info:
5815     \__xeCJK_set_sub_block_family:
5816     \group_end:
5817 }
5818 \tl_new:N \l__xeCJK_family_name_tl
5819 \tl_new:N \l__xeCJK_font_name_tl
5820 \clist_new:N \l__xeCJK_font_options_clist
5821 \cs_generate_variant:Nn \xeCJK_set_family:nnn { e , o }

```

```

\__xeCJK_binding_sub_family: 5822 \cs_new_protected:Npn \__xeCJK_binding_sub_family:
5823 {
5824     \tl_if_empty:NF \l__xeCJK_sub_family_name_tl
5825     { \tl_set_eq:NN \l__xeCJK_family_name_tl \l__xeCJK_sub_family_name_tl }
5826 }

```

```

\__xeCJK_gset_family_cs:n 5827 \cs_new_protected:Npn \__xeCJK_gset_family_cs:n #1
5828 {
5829     \cs_gset_protected:cpe { \__xeCJK_family_csname:n {#1} }
5830     {
5831         \group_begin:
5832         \exp_not:n { \cs_set_eq:NN \xeCJK@fontfamily \use_none:n }
5833         \exp_not:n { \fontspec_gset_family:Nnn \g__xeCJK_fontspec_family_tl }
5834         { \exp_not:o \l__xeCJK_fontspec_options_clist }
5835         { \exp_not:o \l__xeCJK_font_name_tl }
5836         \__xeCJK_gset_family_nfss_cs:no
5837         {#1} { \exp_not:N \g__xeCJK_fontspec_family_tl }

```

```

5838         \group_end:
5839         \tl_set_eq:NN \exp_not:N \l__xeCJK_fontspec_family_tl
5840         \exp_not:N \g__xeCJK_fontspec_family_tl
5841     }
5842 }
5843 \tl_new:N \g__xeCJK_fontspec_family_tl
5844 \tl_new:N \l__xeCJK_fontspec_family_tl

\__xeCJK_check_family:n 5845 \cs_new_protected:Npn \__xeCJK_check_family:n #1
5846 {
5847     \prop_gpop:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
5848     {
5849         \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
5850         {
5851             \cs_undefine:c { \__xeCJK_family_csname:n {#1} }
5852             \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} }
5853         }
5854         \__xeCJK_warning:nee { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
5855     }
5856 }
5857 \cs_generate_variant:Nn \__xeCJK_check_family:n { o }
5858 \__xeCJK_msg_new:nn { CJKfamily-redef }
5859 { Redefining~CJKfamily~\__xeCJK_msg_family_map:n {#1}'~(#2). }

\__xeCJK_parse_font_shape: 5860 \cs_new_protected:Npn \__xeCJK_parse_font_shape:
5861 {
5862     \tl_if_blank:oTF \l__xeCJK_font_name_bf_tl
5863     {
5864         \bool_if:NT \l__xeCJK_auto_fake_bold_bool
5865         {
5866             \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5867             { AutoFakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
5868         }
5869     }
5870     {
5871         \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5872         { BoldFont = { \exp_not:o \l__xeCJK_font_name_bf_tl } }
5873     }
5874     \tl_if_blank:oTF \l__xeCJK_font_name_it_tl
5875     {
5876         \bool_if:NT \l__xeCJK_auto_fake_slant_bool
5877         {
5878             \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5879             { AutoFakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
5880         }
5881     }
5882     {
5883         \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5884         { ItalicFont = { \exp_not:o \l__xeCJK_font_name_it_tl } }
5885     }
5886 }

\g__xeCJK_family_name_prop 5887 \prop_new:N \g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop 5888 \prop_new:N \g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop 5889 \prop_new:N \g__xeCJK_family_font_options_prop

\__xeCJK_save_family_info: 5890 \cs_new_protected:Npn \__xeCJK_save_family_info:
5891 {
5892     \exp_args:Nooo \__xeCJK_save_family_info:nnn
5893     \l__xeCJK_family_name_tl
5894     \l__xeCJK_font_name_tl
5895     \l__xeCJK_font_options_clist
5896 }
5897 \cs_new_protected:Npn \__xeCJK_save_family_info:nnn #1#2#3
5898 {
5899     \prop_gput:Nnn \g__xeCJK_family_font_name_prop {#1} {#2}
5900     \prop_gput:Nnn \g__xeCJK_family_font_options_prop {#1} {#3}
5901 }

```

```

\__xeCJK_set_sub_block_family: 5902 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:
5903 {
5904   \prop_if_empty:NF \l__xeCJK_sub_key_prop
5905   {
5906     \prop_map_function:NN
5907     \l__xeCJK_sub_key_prop
5908     \__xeCJK_set_sub_block_family:nn
5909   }
5910 }
5911 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:nn #1#2
5912 {
5913   \tl_set:Nx \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/#1 }
5914   \quark_if_no_value:nTF {#2}
5915   { \__xeCJK_copy_sub_family:n {#1} }
5916   { \xeCJK_set_family:onn \l__xeCJK_sub_family_name_tl #2 }
5917 }
5918 \cs_new_protected:Npn \__xeCJK_copy_sub_family:n #1
5919 {
5920   \__xeCJK_check_family:o \l__xeCJK_sub_family_name_tl
5921   \prop_get:NnNT \g__xeCJK_family_font_name_prop
5922   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
5923   {
5924     \prop_gput:Noo \g__xeCJK_family_font_name_prop
5925     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
5926   }
5927   \prop_get:NnNT \g__xeCJK_family_font_options_prop
5928   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
5929   {
5930     \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
5931     \prop_gput:Noo \g__xeCJK_family_font_options_prop
5932     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist
5933   }
5934   \cs_gset_protected:cpe
5935   { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
5936   {
5937     \xeCJK_family_if_exist:eT { \l__xeCJK_family_name_tl }
5938     {
5939       \__xeCJK_gset_family_nfss_cs:no
5940       { \l__xeCJK_sub_family_name_tl }
5941       { \exp_not:N \l__xeCJK_fontspec_family_tl }
5942     }
5943   }
5944 }
\__xeCJK_copy_family:nn 5945 \cs_new_protected:Npn \__xeCJK_copy_family:nn #1#2
\__xeCJK_copy_family:ee 5946 {
5947   \xeCJK_family_if_exist:nT {#2}
5948   {
5949     \prop_gput:Nno \g__xeCJK_family_name_prop
5950     {#1} \l__xeCJK_fontspec_family_tl
5951     \tl_map_inline:nn
5952     {
5953       \g__xeCJK_family_font_name_prop
5954       \g__xeCJK_family_font_options_prop
5955     }
5956     {
5957       \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
5958       { \prop_gput:Nno ##1 {#1} \l__xeCJK_tmp_tl }
5959     }
5960     \cs_gset_eq:cc
5961     { \__xeCJK_family_nfss_csname:n {#1} }
5962     { \__xeCJK_family_nfss_csname:n {#2} }
5963   }
5964 }
5965 \cs_generate_variant:Nn \__xeCJK_copy_family:nn { ee }

```

5.13 字体切换

\xeCJK_select_font: 缓存当前字体的原始格式, 以加速编译。

```

\l_xeCJK_current_font_tl
5966 \cs_new:Npn \__xeCJK_font_csname:n #1
5967 { xeCJK/#1/\f@series/\f@shape/\f@size }
5968 \tl_new:N \l_xeCJK_current_font_tl
5969 \tl_set:No \l_xeCJK_current_font_tl
5970 { \__xeCJK_font_csname:n { \CJK@family } }
5971 \cs_new_protected:Npn \xeCJK_select_font:
5972 {
5973   \__xeCJK_select_font:cn
5974   { \l_xeCJK_current_font_tl }
5975   { \l_xeCJK_family_tl }
5976 }
5977 \cs_new_protected:Npn \__xeCJK_select_font:Nn #1#2
5978 {
5979   \xeCJK_clear_fallback_font:
5980   \cs_if_exist:NF #1 { \__xeCJK_font_initial:Nn #1 {#2} }
5981   #1
5982 }
5983 \cs_generate_variant:Nn \__xeCJK_select_font:Nn { c }
5984 \tl_new:N \l__xeCJK_current_coord_tl
5985 \cs_new_eq:NN \xeCJK_setfont \xeCJK_select_font:

```

__xeCJK_font_initial:Nn 注意要将 \selectfont 放在分组中调用, 防止 \f@series 等字体参数被修改, 导致 \l_xeCJK_current_font_tl 标记前后不一致, 引发错误(见 #486)。

```

5986 \cs_new_protected:Npn \__xeCJK_font_initial:Nn #1#2
5987 {
5988   \group_begin:
5989   \__xeCJK_font_initial_hook:
5990   \__xeCJK_family_use:n {#2}
5991   \xeCJK_font_gset_to_current:N #1
5992   \group_end:
5993 }
5994 \cs_new_protected:Npn \__xeCJK_font_initial_hook:
5995 { \tl_use:N \g__xeCJK_font_initial_hook_tl }
5996 \cs_new_protected:Npn \__xeCJK_gadd_font_initial_hook:n
5997 { \tl_gput_right:Nn \g__xeCJK_font_initial_hook_tl }
5998 \tl_new:N \g__xeCJK_font_initial_hook_tl

```

\xeCJK_select_punct_font: 切换标点符号字体。

```

\l_xeCJK_current_punct_font_tl
5999 \cs_new_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
6000 \cs_new_protected:Npn \__xeCJK_select_punct_font_aux:
6001 {
6002   \__xeCJK_select_font:cn
6003   { \l_xeCJK_current_punct_font_tl }
6004   { \l_xeCJK_punct_family_tl }
6005 }
6006 \tl_new:N \CJK@punctfamily
6007 \tl_new:N \l_xeCJK_punct_family_tl
6008 \tl_new:N \l_xeCJK_current_punct_font_tl
6009 \tl_set:Nn \CJK@punctfamily { \CJK@family }
6010 \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
6011 \tl_set:No \l_xeCJK_current_punct_font_tl
6012 { \__xeCJK_font_csname:n { \CJK@punctfamily } }
6013 \cs_new_eq:NN \__xeCJK_select_font: \prg_do_nothing:
6014 \cs_new_eq:NN \__xeCJK_select_punct_font: \prg_do_nothing:

```

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

```

6015 \cs_new_protected:Npn \__xeCJK_switch_font:nn #1#2
6016 {
6017   \str_if_eq:nnF {#1} {#2}
6018   {

```

```

6019     \__xeCJK_info:nee { CJK-block } {#1} {#2}
6020     \str_if_eq:nnTF {#2} { CJK }
6021     { \xeCJK_select_font: }
6022     { \xeCJK_select_font:n {#2} }
6023   }
6024 }
6025 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~`#1'~to~`#2'. }

```

\xeCJK_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;
 \xeCJK_block_family:nn 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。

```

6026 \cs_new_protected:Npn \xeCJK_select_font:n #1
6027 {
6028   \__xeCJK_select_font:cnn
6029   { \__xeCJK_font_csname:n { \CJK@family/#1 } }
6030   { \l_xeCJK_family_tl }
6031   {#1}
6032 }
6033 \cs_new_protected:Npn \__xeCJK_select_font:Nnn #1#2#3
6034 {
6035   \xeCJK_clear_fallback_font:
6036   \cs_if_exist:NF #1
6037   { \__xeCJK_block_font_initial:Nnn #1 {#2} {#3} }
6038   #1
6039 }
6040 \cs_generate_variant:Nn \__xeCJK_select_font:Nnn { c }
6041 \cs_new_protected:Npn \__xeCJK_block_font_initial:Nnn #1#2#3
6042 {
6043   \xeCJK_block_family:nn {#2} {#3}
6044   \__xeCJK_font_initial:Nn #1 { #2/#3 }
6045 }
6046 \cs_new_protected:Npn \xeCJK_block_family:nn #1#2
6047 {
6048   \xeCJK_family_if_exist:eF { #1/#2 }
6049   {
6050     \__xeCJK_copy_family:ee { #1/#2 }
6051     {
6052       \cs_if_exist:cTF
6053       { \__xeCJK_family_csname:n { \CJKfamilydefault/#2 } }
6054       { \CJKfamilydefault/#2 } {#1}
6055     }
6056   }
6057 }

```

```

\__xeCJK_family_csname:n 6058 \cs_new:Npn \__xeCJK_family_csname:n #1
\__xeCJK_family_nfss_csname:n 6059 { xeCJK/family/#1 }
\__xeCJK_family_use:n 6060 \cs_new:Npn \__xeCJK_family_nfss_csname:n #1
\__xeCJK_gset_family_nfss_cs:nn 6061 { xeCJK/family/nfss/#1 }
6062 \cs_new_protected:Npn \__xeCJK_family_use:n #1
6063 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
6064 \cs_new_protected:Npn \__xeCJK_gset_family_nfss_cs:nn #1#2
6065 {
6066   \prop_gput:Nnn \g__xeCJK_family_name_prop {#1} {#2}
6067   \cs_gset_protected:cpe
6068   { \__xeCJK_family_nfss_csname:n {#1} }
6069   { \__xeCJK_nfss_family:nn { \c__xeCJK_encoding_tl } {#2} }
6070 }
6071 \cs_generate_variant:Nn \__xeCJK_gset_family_nfss_cs:nn { no }

```

__xeCJK_nfss_family:n 用于处理 L^AT_EX 2_ε 2020/02/02 中 \bfseries@rm 等与 \bfdefault 不一致可能导致的问题。

```

6072 \cs_if_exist:NTF \fontseriesforce
6073 {
6074   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
6075   {
6076     \fontencoding {#1}
6077     \str_if_eq:eeF { \f@series } { \bfdefault }

```

```

6078     {
6079         \str_case:e:nn { \f@family }
6080         {
6081             { \rmdefault } { \_xeCJK_nfss_series:n { rm } }
6082             { \sfdefault } { \_xeCJK_nfss_series:n { sf } }
6083             { \ttdefault } { \_xeCJK_nfss_series:n { tt } }
6084         }
6085     }
6086     \fontfamily {#2}
6087     \selectfont
6088 }
6089 \cs_new_protected:Npn \_xeCJK_nfss_series:n #1
6090 {
6091     \str_if_eq:eeT { \f@series } { \use:c { bfseries@#1 } }
6092     { \fontseriesforce { \bfdefault } }
6093 }
6094 }
6095 {
6096     \cs_new_protected:Npn \_xeCJK_nfss_family:nn #1#2
6097     {
6098         \fontencoding {#1}
6099         \tl_set:Nn \f@family {#2}
6100         \selectfont
6101     }
6102 }

```

```

\xeCJK_family_if_exist:nTF 6103 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:n #1 { T , F , TF }
6104 {
6105     \prop_get:NnNTF \g__xeCJK_family_name_prop
6106     {#1} \l__xeCJK_fontspec_family_tl
6107     { \prg_return_true: }
6108     {
6109         \exp_args:Ne \cs_if_exist_use:cTF
6110         { \_xeCJK_family_csname:n {#1} }
6111         { \prg_return_true: }
6112         { \prg_return_false: }
6113     }
6114 }
6115 \prg_generate_conditional_variant:Nnn \xeCJK_family_if_exist:n { e } { T , F , TF }

```

\CJKfamily 用于切换 CJK 字体族。

```

6116 \NewDocumentCommand \CJKfamily { t+ t- m }
6117 {
6118     \xeCJK_family:NNe #1 #2 {#3}
6119     \tex_ignorespaces:D
6120 }
6121 \cs_new_protected:Npn \xeCJK_family:NNn #1#2#3
6122 {
6123     \tl_if_blank:nTF {#3}
6124     {
6125         \bool_if:NF #1 { \bool_if:NF #2 { \use_none:nn } }
6126         \xeCJK_family_if_exist_use:e { \l__xeCJK_family_tl }
6127     }
6128     {
6129         \bool_if:NTF #2
6130         { \xeCJK_family_if_exist_use:n {#3} }
6131         {
6132             \xeCJK_family_if_exist:nTF {#3}
6133             {
6134                 \tl_set:Nn \l__xeCJK_family_tl {#3}
6135                 \tl_set_eq:NN \CJK@family \l__xeCJK_fontspec_family_tl
6136                 \bool_if:NT #1 { \_xeCJK_family_use:n {#3} }
6137             }
6138             { \_xeCJK_family_unknown_warning:n {#3} }
6139         }
6140     }
6141 }

```

```

6142 \cs_generate_variant:Nn \xeCJK_family:NNn { NNe }
6143 \cs_new_protected:Npn \xeCJK_switch_family:n #1
6144 {
6145   \xeCJK_family_if_exist:nTF {#1}
6146   {
6147     \tl_set:Nn \l_xeCJK_family_tl {#1}
6148     \tl_set_eq:NN \CJK@family \l__xeCJK_fontspec_family_tl
6149   }
6150   { \__xeCJK_family_unknown_warning:n {#1} }
6151 }
6152 \cs_generate_variant:Nn \xeCJK_switch_family:n { e , o }

```

PunctFamily 设置汉字标点符号的字体。

```

6153 \keys_define:nn { xeCJK / options }
6154 {
6155   PunctFamily .choice: ,
6156   PunctFamily .value_required:n = { true } ,
6157   PunctFamily / false .code:n =
6158   {
6159     \tl_set:Nn \CJK@punctfamily { \CJK@family }
6160     \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
6161     \xeCJK_cs_clear:N \__xeCJK_select_font:
6162     \xeCJK_cs_clear:N \__xeCJK_select_punct_font:
6163     \cs_set_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
6164   } ,
6165   PunctFamily / unknown .code:n =
6166   { \xeCJK_punct_family:e {#1} } ,
6167 }
6168 \cs_new_protected:Npn \xeCJK_punct_family:n #1
6169 {
6170   \xeCJK_family_if_exist:nTF {#1}
6171   {
6172     \tl_set:Nn \l_xeCJK_punct_family_tl {#1}
6173     \tl_set_eq:NN \CJK@punctfamily \l__xeCJK_fontspec_family_tl
6174     \cs_set_eq:NN \__xeCJK_select_font: \xeCJK_select_font:
6175     \cs_set_eq:NN \__xeCJK_select_punct_font: \__xeCJK_select_punct_font_aux:
6176     \cs_set_eq:NN \xeCJK_select_punct_font: \__xeCJK_select_punct_font:
6177   }
6178   { \__xeCJK_family_unknown_warning:n {#1} }
6179 }
6180 \cs_generate_variant:Nn \xeCJK_punct_family:n { e }

```

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

```
6181 \tl_new:N \l_xeCJK_family_tl
```

`\CJK@family` 用于保存实际的字体族名称。

```
6182 \tl_new:N \CJK@family
```

```

\__xeCJK_gobble_CJKfamily: 6183 \cs_new_protected:Npn \__xeCJK_gobble_CJKfamily:
6184 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
6185 \NewExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }

\xeCJK_family_if_exist_use:n 6186 \cs_new_protected:Npn \xeCJK_family_if_exist_use:n #1
6187 {
6188   \xeCJK_family_if_exist:nTF {#1}
6189   { \__xeCJK_family_use:n {#1} }
6190   { \__xeCJK_family_unknown_warning:n {#1} }
6191 }
6192 \cs_generate_variant:Nn \xeCJK_family_if_exist_use:n { e }

\__xeCJK_family_unknown_warning:n 6193 \cs_new_protected:Npn \__xeCJK_family_unknown_warning:n #1
6194 {
6195   \prop_if_empty:NF \g__xeCJK_family_font_name_prop
6196   {
6197     \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}

```

```

6198         {
6199             \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
6200             \__xeCJK_warning:ne { CJKfamily-Unknown } {#1}
6201         }
6202     }
6203 }
6204 \seq_new:N \g__xeCJK_unknown_family_seq
6205 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
6206 {
6207     Unknown~CJK~family~\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
6208     Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
6209 }
6210 \cs_new:Npn \__xeCJK_msg_def_family_map:n #1
6211 {
6212     \str_case:e:nnF {#1}
6213     {
6214         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
6215         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
6216         \CJKttdefault { \token_to_str:N \setCJKmonofont }
6217     }
6218     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
6219     [<...>]\{<...>\}
6220 }
6221 \cs_new:Npn \__xeCJK_msg_family_map:n #1
6222 {
6223     \str_case:e:nnF {#1}
6224     {
6225         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
6226         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
6227         \CJKttdefault { \token_to_str:N \CJKttdefault }
6228     }
6229     {#1}
6230 }

```

__xeCJK_pass_args:nnnn 为了支持字体属性可选项在前和在后两种语法, 给出两个辅助工具, 类似 **fontspec** 的实现。自带展开功能, 额外参数 #4 用于后处理。

```

6231 \cs_new_protected:Npn \__xeCJK_pass_args:nnnn #1#2#3#4
6232 {
6233     \tl_if_novalue:nTF {#2}
6234     { \__xeCJK_post_arg:w {#1} {#3} {#4} }
6235     {
6236         \use:e { #1 {#2} {#3} }
6237         #4
6238     }
6239 }
6240 \NewDocumentCommand \__xeCJK_post_arg:w { m m m O { } }
6241 {
6242     \use:e { #1 {#4} {#2} }
6243     #3
6244 }

```

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

```

\setCJKsansfont
\setCJKmonofont
6245 \NewDocumentCommand \setCJKmainfont { o m }
6246 {
6247     \__xeCJK_pass_args:nnnn
6248     { \xeCJK_set_family:nnn { \CJKrmdefault } } {#1} {#2}
6249     { \__xeCJK_preamble_family:n { \CJKrmdefault } }
6250 }
6251 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
6252 \NewDocumentCommand \setCJKsansfont { o m }
6253 {
6254     \__xeCJK_pass_args:nnnn
6255     { \xeCJK_set_family:nnn { \CJKsfdefault } } {#1} {#2}
6256     { \__xeCJK_preamble_family:n { \CJKsfdefault } }
6257 }
6258 \NewDocumentCommand \setCJKmonofont { o m }

```

```

6259 {
6260   \__xeCJK_pass_args:nnnn
6261   { \xeCJK_set_family:nnn { \CJKttdefault } } {#1} {#2}
6262   { \__xeCJK_preamble_family:n { \CJKttdefault } }
6263 }

6264 \@onlypreamble \setCJKmainfont
6265 \@onlypreamble \setCJKmathfont
6266 \@onlypreamble \setCJKsansfont
6267 \@onlypreamble \setCJKmonofont
6268 \@onlypreamble \setCJKromanfont

```

__xeCJK_preamble_family:n 用在 \setCJKmainfont 等主要命令之后, 确保导言区有 CJK 字体可用。

```

6269 \cs_new_protected:Npn \__xeCJK_preamble_family:n #1
6270 { \str_if_eq:eeT {#1} { \CJKfamilydefault } { \normalfont } }

```

\setCJKfamilyfont 分别用于预声明 CJK 字体族和声明并马上调用 CJK 字体族。
 \newCJKfontfamily
 \CJKfontspec

```

6271 \NewDocumentCommand \setCJKfamilyfont { m o m }
6272 {
6273   \__xeCJK_pass_args:nnnn
6274   { \xeCJK_set_family:nnn {#1} } {#2} {#3}
6275   { }
6276 }
6277 \__xeCJK_msg_new:nn { command-already-defined }
6278 { Control~sequence~#1~is~already~defined. }
6279 \NewDocumentCommand \newCJKfontfamily { o m o m }
6280 {
6281   \tl_set:Nx \l__xeCJK_tmp_tl
6282   { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
6283   \cs_if_exist:NTF #2
6284   { \__xeCJK_error:ne { command-already-defined } { \token_to_str:N #2 } }
6285   {
6286     \cs_set_protected:Npe #2
6287     { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
6288   }
6289   \__xeCJK_pass_args:nnnn
6290   { \xeCJK_set_family:nnn { \l__xeCJK_tmp_tl } } {#3} {#4}
6291   { }
6292 }
6293 \NewDocumentCommand \CJKfontspec { o m }
6294 {
6295   \__xeCJK_pass_args:nnnn
6296   { \xeCJK_fontspec:nn } {#1} {#2}
6297   { \tex_ignorespaces:D }
6298 }

```

```

\xeCJK_fontspec:nn 6299 \cs_new_protected:Npn \xeCJK_fontspec:nn #1#2
6300 {
6301   \prop_get:NnNTF \g__xeCJK_fontspec_prop
6302   { CJKfontspec/#1/#2/id } \l__xeCJK_family_tl
6303   { \xeCJK_switch_family:o { \l__xeCJK_family_tl } }
6304   {
6305     \__xeCJK_fontspec:enn
6306     { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + 1 } ) }
6307     {#1} {#2}
6308   }
6309 }
6310 \cs_new_protected:Npn \__xeCJK_fontspec:nnn #1#2#3
6311 {
6312   \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
6313   \xeCJK_set_family:nnn {#1} {#2} {#3}
6314   \xeCJK_switch_family:n {#1}
6315 }
6316 \cs_generate_variant:Nn \xeCJK_fontspec:nn { oo }
6317 \cs_generate_variant:Nn \__xeCJK_fontspec:nnn { e }
6318 \prop_new:N \g__xeCJK_fontspec_prop

```

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。

```

\defaultCJKfontfeatures
\addCJKfontfeatures
6319 \clist_new:N \g__xeCJK_default_features_clist
6320 \NewDocumentCommand \defaultCJKfontfeatures { m }
6321 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
6322 \@onlypreamble \defaultCJKfontfeatures
6323 \NewDocumentCommand \addCJKfontfeatures { s O { } m }
6324 {
6325   \xeCJK_add_font_features:Nee #1 {#2} {#3}
6326   \tex_ignorespaces:D
6327 }
6328 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\xeCJK_add_font_features:Nnn 6329 \cs_new_protected:Npn \xeCJK_add_font_features:Nnn #1#2#3
6330 {
6331   \prop_get:NoNTF \g__xeCJK_family_font_name_prop
6332   \l_xeCJK_family_tl \l__xeCJK_font_name_tl
6333   {
6334     \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
6335     \seq_map_inline:Nn \g__xeCJK_sub_key_seq
6336     { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
6337     \seq_clear:N \l__xeCJK_sub_key_seq
6338     \clist_clear:N \l__xeCJK_add_block_features_clist
6339     \clist_map_function:nN {#2} \__xeCJK_add_sub_features:n
6340     \bool_lazy_and:nnT
6341     {#1}
6342     { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
6343     {
6344       \seq_map_function:NN
6345       \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
6346     }
6347     \prop_get:NoNT \g__xeCJK_family_font_options_prop
6348     \l_xeCJK_family_tl \l__xeCJK_font_options_clist
6349     {
6350       \bool_lazy_or:nnT
6351       { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
6352       {#1}
6353       {
6354         \clist_concat:NNN \l__xeCJK_font_options_clist
6355         \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
6356       }
6357       \clist_concat:NNN \l__xeCJK_font_options_clist
6358       \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
6359     }
6360     \xeCJK_fontspec:oo \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
6361   }
6362   { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
6363 }
6364 \cs_new_protected:Npn \__xeCJK_add_sub_features:n #1
6365 {
6366   \seq_if_in:NnTF \g__xeCJK_sub_key_seq {#1}
6367   {
6368     \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
6369     \__xeCJK_add_sub_class_features:n {#1}
6370   }
6371   { \__xeCJK_warning:ne { SubBlock-undefined } {#1} }
6372 }
6373 \clist_new:N \l__xeCJK_add_font_features_clist
6374 \clist_new:N \l__xeCJK_add_block_features_clist
6375 \cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nee , Nne }
6376 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
6377 {
6378   \token_to_str:N \addCJKfontfeature (s)~ignored.\\
6379   It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
6380 }

\__xeCJK_add_sub_class_features:n 6381 \cs_new_protected:Npn \__xeCJK_add_sub_class_features:n #1
6382 {

```

```

6383 \prop_get:NoNTF \g__xeCJK_family_font_name_prop
6384 { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
6385 {
6386   \prop_get:NoN \g__xeCJK_family_font_options_prop
6387   { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
6388 }
6389 {
6390   \prop_get:NeNTF \g__xeCJK_family_font_name_prop
6391   { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
6392   {
6393     \prop_get:NeN \g__xeCJK_family_font_options_prop
6394     { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
6395   }
6396   {
6397     \prop_get:NoN \g__xeCJK_family_font_options_prop
6398     \l_xeCJK_family_tl \l__xeCJK_sub_font_options_clist
6399     \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl
6400   }
6401 }
6402 \clist_concat:NNN \l__xeCJK_sub_font_options_clist
6403 \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
6404 \clist_put_right:Ne \l__xeCJK_add_block_features_clist
6405 {
6406   #1 =
6407   {
6408     [ \exp_not:o \l__xeCJK_sub_font_options_clist ]
6409     { \exp_not:o \l__xeCJK_sub_font_name_tl }
6410   }
6411 }
6412 }
6413 \cs_generate_variant:Nn \prop_get:NnN { Ne }
6414 \prg_generate_conditional_variant:Nnn \prop_get:NnN { Ne } { T , TF }

LoadFandol 6415 \keys_define:nn { xeCJK / options }
6416 { LoadFandol .bool_gset:N = \g__xeCJK_fandol_bool }
6417 \cs_new_protected:Npn \__xeCJK_load_fandol:
6418 {
6419   \xeCJK_set_family:enn { \CJKrmdefault }
6420   { Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular }
6421   { FandolSong-Regular }
6422   \xeCJK_set_family:enn { \CJKsfdefault }
6423   { Extension = .otf , BoldFont = FandolHei-Bold } { FandolHei-Regular }
6424   \xeCJK_set_family:enn { \CJKttdefault }
6425   { Extension = .otf } { FandolFang-Regular }
6426 }

```

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

```

6427 \__xeCJK_at_end_preamble:n
6428 {
6429   \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
6430   {
6431     \group_begin:
6432     \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
6433     \tl_gset:Ne \CJKfamilydefault
6434     {
6435       \str_case:onF { \familydefault }
6436       {
6437         { \rmdefault } { \exp_not:N \CJKrmdefault }
6438         { \sfdefault } { \exp_not:N \CJKsfdefault }
6439         { \ttdefault } { \exp_not:N \CJKttdefault }
6440       }
6441       { \CJKfamilydefault }
6442     }

```

```

6443     \group_end:
6444   }
6445   \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
6446   {
6447     \bool_if:NTF \g__xeCJK_fandol_bool
6448     {
6449       \__xeCJK_warning:n { fandol }
6450       \__xeCJK_load_fandol:
6451       \xeCJK_ensure_default_family:
6452     }
6453     { \__xeCJK_warning:ne { no-CJKfamily } { \CJKfamilydefault } }
6454   }
6455   { \xeCJK_ensure_default_family: }
6456 }
6457 \cs_new_protected:Npn \xeCJK_ensure_default_family:
6458 {
6459   \xeCJK_family_if_exist:eF { \CJKfamilydefault }
6460   {
6461     \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
6462     \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
6463     { \use:n }
6464     {
6465       \xeCJK_family_if_exist:eTF { \CJKrmdefault }
6466       { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
6467     }
6468     {
6469       \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
6470       {
6471         \prop_map_break:n
6472         { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
6473       }
6474     }
6475     \__xeCJK_warning:nee { CJKfamilydefault-undefined }
6476     { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
6477   }
6478   \xeCJK_switch_family:e { \CJKfamilydefault }
6479   \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
6480 }
6481 \__xeCJK_msg_new:nn { no-CJKfamily }
6482 {
6483   It~seems~that~you~have~not~declared~a~CJKfamily.\\
6484   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
6485   \__xeCJK_msg_def_family_map:n {#1}'\\
6486   in~the~preamble~to~declare~the~default~CJKfamily.\\
6487 }
6488 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
6489 {
6490   Undefined~CJK~default~family~\__xeCJK_msg_family_map:n {#1}'~
6491   has~been~replaced~by~\__xeCJK_msg_family_map:n {#2}'\\.\\
6492   Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
6493 }
6494 \__xeCJK_msg_new:nn { fandol }
6495 {
6496   Fandol~is~being~set~as~the~default~font~for~CJK~text.\\
6497   Please~make~sure~it~has~been~properly~installed.
6498 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```
6499 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }
```

\setCJKmathfont 设置 CJK 数学字体。

```
6500 \NewDocumentCommand \setCJKmathfont { o m }
6501 {
```

```

6502     \__xeCJK_pass_args:nnnn
6503     { \xeCJK_set_family:nnn { \c__xeCJK_math_tl } } {#1} {#2}
6504     { }
6505 }
6506 \tl_const:Nn \c__xeCJK_math_tl { CJKmath }

```

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

```

6507 \cs_new_protected:Npn \xeCJK_set_mathfont:
6508 {
6509     \cs_if_exist_use:N \__xeCJK_save_um_char:
6510     \xeCJK_family_if_exist:ETF { \c__xeCJK_math_tl }
6511     { \__xeCJK_set_mathfont_aux: }
6512     {
6513         \xeCJK_family_if_exist:ET { \CJKfamilydefault }
6514         {
6515             \__xeCJK_copy_family:ee { \c__xeCJK_math_tl } { \CJKfamilydefault }
6516             \__xeCJK_set_mathfont_aux:
6517         }
6518     }
6519     \cs_if_exist_use:N \__xeCJK_restore_um_char:
6520 }
6521 \cs_new_protected:Npn \__xeCJK_set_mathfont_aux:
6522 {
6523     \tl_const:Nn \c__xeCJK_math_family_tl
6524     { \l__xeCJK_fontspec_family_tl }
6525     \xeCJK_declare_mathfont:ee
6526     { \c__xeCJK_math_tl }
6527     { \c__xeCJK_math_family_tl }
6528     \int_const:Nn \c__xeCJK_math_fam_int
6529     { \use:c { sym \c__xeCJK_math_tl } }
6530     \clist_gconcat:NNN \g__xeCJK_math_chars_clist
6531     \g__xeCJK_CJK_range_clist \g__xeCJK_FullLeft_range_clist
6532     \clist_gconcat:NNN \g__xeCJK_math_chars_clist
6533     \g__xeCJK_math_chars_clist \g__xeCJK_FullRight_range_clist
6534     \xeCJK_gset_mathcode:Nn \g__xeCJK_math_chars_clist
6535     { \c__xeCJK_math_fam_int }
6536     \xeCJK_set_mathfont_block:
6537 }
6538 \clist_new:N \g__xeCJK_math_chars_clist
6539 \prop_new:N \g__xeCJK_fam_prop

```

\xeCJK_set_mathfont_block: 分区数学字体。

```

6540 \cs_new_protected:Npn \xeCJK_set_mathfont_block:
6541 {
6542     \seq_if_empty:NF \g__xeCJK_CJK_sub_class_seq
6543     {
6544         \seq_map_function:NN
6545         \g__xeCJK_CJK_sub_class_seq
6546         \xeCJK_set_mathfont_block:n
6547     }
6548 }
6549 \cs_new_protected:Npn \xeCJK_set_mathfont_block:n #1
6550 {
6551     \xeCJK_block_family:nn { \c__xeCJK_math_tl } {#1}
6552     \prop_get:NoNTF \g__xeCJK_fam_prop
6553     \l__xeCJK_fontspec_family_tl \l__xeCJK_tmp_tl
6554     { \int_set:Nn \l__xeCJK_fam_int { \l__xeCJK_tmp_tl } }
6555     {
6556         \xeCJK_declare_mathfont:ee
6557         { \c__xeCJK_math_tl / #1 }
6558         { \l__xeCJK_fontspec_family_tl }
6559         \__xeCJK_set_mathfont_block_aux:cn
6560         { sym \c__xeCJK_math_tl / #1 } {#1}
6561     }
6562     \xeCJK_gset_mathcode:cn { g__xeCJK_CJK/#1_range_clist } { \l__xeCJK_fam_int }

```

```

6563 }
6564 \cs_new_protected:Npn \__xeCJK_set_mathfont_block_aux:Nn #1#2
6565 {
6566   \int_set_eq:NN \l__xeCJK_fam_int #1
6567   \prop_gput:Nnn \g__xeCJK_block_fam_prop {#2} {#1}
6568 }
6569 \int_new:N \l__xeCJK_fam_int
6570 \prop_new:N \g__xeCJK_block_fam_prop
6571 \cs_generate_variant:Nn \__xeCJK_set_mathfont_block_aux:Nn { c }

```

`\xeCJK_declare_mathfont:nn` 注意从 L^AT_EX 2_ε 2020/02/02 开始, `\shapedefault` 初始值是 `n`, 而 `\updefault` 初始值是 `up`, 两者并不一致。fontspec 包定义字体使用的是 `\shapedefault`。

```

6572 \cs_new_protected:Npn \xeCJK_declare_mathfont:nn #1#2
6573 {
6574   \xeCJK_declare_symbol_font:nnnnn {#1} { \c__xeCJK_encoding_tl }
6575   {#2} { \mddefault } { \shapedefault }
6576   \cs_if_free:cF
6577   { \c__xeCJK_encoding_tl/#2/\bfdefault/\shapedefault }
6578   {
6579     \SetSymbolFont {#1} { bold } { \c__xeCJK_encoding_tl }
6580     {#2} { \bfdefault } { \shapedefault }
6581   }
6582   \prop_gput:Nne \g__xeCJK_fam_prop {#2} { \exp_not:c { sym #1 } }
6583 }
6584 \cs_generate_variant:Nn \prop_put:Nnn { Nne }
6585 \cs_generate_variant:Nn \prop_gput:Nnn { Nne }
6586 \cs_generate_variant:Nn \xeCJK_declare_mathfont:nn { ee }

```

`\xeCJK_declare_symbol_font:nnnnn` 主要功能同 `\DeclareSymbolFont`, 不带编码和重复定义检查。

```

6587 \cs_new_protected:Npn \xeCJK_declare_symbol_font:nnnnn #1
6588 { \__xeCJK_declare_symbol_font:cnnnn { sym #1 } }
6589 \cs_new_protected:Npn \__xeCJK_declare_symbol_font:Nnnnn #1
6590 {
6591   \xeCJK_new_fam:N #1
6592   \xeCJK_new_symbol_font:Nnnnn #1
6593 }
6594 \cs_generate_variant:Nn \__xeCJK_declare_symbol_font:Nnnnn { c }

```

`\xeCJK_new_fam:N` 我们从 255 往下分配 `\fam`, `\count18` 是 L^AT_EX 2_ε 记录最后分配的 `\fam` 编号, 作为我们的分配器的下限。事实上, 还应该相应地减小 `\e@mathgroup@top` 才合理, 但这可能会有不利影响, 我们暂未处理。

```

6595 \cs_new_protected:Npn \xeCJK_new_fam:N #1
6596 {
6597   \int_compare:nNtF
6598   { \g__xeCJK_fam_allocation_int } > { \g__xeCJK_fam_bottom_int }
6599   {
6600     \int_set_eq:NN \allocationnumber \g__xeCJK_fam_allocation_int
6601     \int_const:Nn #1 { \allocationnumber }
6602     \iow_log:e
6603     {
6604       \token_to_str:N #1 =
6605       \token_to_str:N \mathgroup \int_use:N \allocationnumber
6606     }
6607     \int_gdecr:N \g__xeCJK_fam_allocation_int
6608   }
6609   { \__xeCJK_error:n { fam-exhausted } }
6610 }
6611 \tex_countdef:D \g__xeCJK_fam_bottom_int = 18 ~
6612 \int_new:N \g__xeCJK_fam_allocation_int
6613 \int_gset:Nn \g__xeCJK_fam_allocation_int { 255 }
6614 \__xeCJK_msg_new:nn { fam-exhausted }
6615 { No~room~for~a~new~fam. }

```

`\xeCJK_new_symbol_font:Nnnnn` 功能同 `\new@symbolfont`, 但我们不增加 `\c@mv@normal` 和 `\c@mv@bold` 之类的计数器。
`\_xeCJK_new_symbol_font:NN`

```

6616 \cs_new_protected:Npn \xeCJK_new_symbol_font:Nnnnn #1#2#3#4#5
6617 { \_xeCJK_new_symbol_font:Nc #1 { #2/#3/#4/#5 } }
6618 \cs_new_protected:Npn \_xeCJK_new_symbol_font:NN #1#2
6619 {
6620   \tl_put_right:Nn \group@list { \group@elt #1 #2 }
6621   \cs_set:Npn \version@elt ##1
6622     { \tl_put_right:Nn ##1 { \getanddefine@fonts #1 #2 } }
6623   \version@list
6624 }
6625 \cs_generate_variant:Nn \_xeCJK_new_symbol_font:NN { Nc }

```

`\xeCJK_gset_mathcode:Nn` CJK 字符的数学类别固定为 0(`\mathord`)。

`\xeCJK_gset_mathcode:Nnn`
`\xeCJK_gset_mathcode:nnnn`

```

6626 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nn #1#2
6627 {
6628   \clist_map_inline:Nn #1
6629   {
6630     \_xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn { ##1 }
6631     { 0 } {#2}
6632   }
6633 }
6634 \cs_generate_variant:Nn \xeCJK_gset_mathcode:Nn { c }
6635 \cs_new_protected:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
6636 {
6637   \_xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
6638   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
6639   {
6640     \xeCJK_gset_mathcode:Nnn \l__xeCJK_begin_int {#3} {#4}
6641     \int_incr:N \l__xeCJK_begin_int
6642   }
6643 }
6644 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nnn #1#2#3
6645 { \tex_global:D \tex_Umathcode:D #1 = #2 ~ #3 ~ #1 }

```

5.15 抄录环境中的间距调整

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

```

6646 \int_new:N \l__xeCJK_verb_case_int
6647 \keys_define:nn { xeCJK / options }
6648 {
6649   Verb .choices:nn =
6650     { true , env+ , env , false }
6651     { \int_set_eq:NN \l__xeCJK_verb_case_int \l_keys_choice_int } ,
6652   Verb .default:n = { env }
6653 }
6654 \cs_new_protected:Npn \_xeCJK_verb_font_hook:
6655 {
6656   \if_case:w \l__xeCJK_verb_case_int
6657   \or:
6658     \_xeCJK_nobreak_skip_zero:
6659   \or:
6660     \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
6661       { \xeCJKVerbAddon }
6662       { \_xeCJK_nobreak_skip: }
6663   \or:
6664     \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
6665       { \xeCJKVerbAddon }
6666       { \_xeCJK_nobreak_skip_zero: }
6667   \fi:
6668 }
6669 \_xeCJK_after_preamble:n
6670 {
6671   \cs_set_protected:Npe \verbatim@font

```

```

6672     { \exp_not:o { \verbatim@font } \__xeCJK_verb_font_hook: }
6673 }

\__xeCJK_nobreak_skip_zero: 6674 \cs_new_protected:Npn \__xeCJK_nobreak_skip_zero:
\__xeCJK_nobreak_skip: 6675 {
6676     \__xeCJK_reset_shipout_skip:
6677     \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
6678     \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
6679     \tl_put_right:Nn \l__xeCJK_reset_shipout_skip_hook_tl
6680     {
6681         \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
6682         \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
6683     }
6684     \xeCJK_cs_clear:N \CJKglue
6685     \xeCJK_cs_clear:N \CJKecglue
6686     \xeCJK_cs_clear:N \xeCJK_check_for_glue:
6687     \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \xeCJK_class_group_end:
6688     \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
6689     \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
6690 }
6691 \cs_new_protected:Npn \__xeCJK_nobreak_skip:
6692 {
6693     \__xeCJK_reset_shipout_skip:
6694     \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_ccglue_skip
6695     \skip_if_eq:nnTF { \l__xeCJK_ccglue_skip } { \c_zero_skip }
6696     { \xeCJK_cs_clear:N \CJKglue }
6697     { \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue: }
6698     \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
6699     \skip_if_eq:nnTF { \l__xeCJK_ecglue_skip } { \c_zero_skip }
6700     { \xeCJK_cs_clear:N \CJKecglue }
6701     { \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue: }
6702     \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
6703     \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
6704 }
6705 \cs_new_protected:Npn \__xeCJK_nobreak_ccglue:
6706 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ccglue_skip }
6707 \cs_new_protected:Npn \__xeCJK_nobreak_ecglue:
6708 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ecglue_skip }

\__xeCJK_reset_shipout_skip: 6709 \cs_new_protected:Npn \__xeCJK_reset_shipout_skip:
6710 {
6711     \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
6712     \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
6713     \cs_set_eq:NN \__xeCJK_shipout_punct_hskip:n \__xeCJK_punct_hskip:n
6714     \cs_set_eq:NN
6715         \__xeCJK_shipout_punct_breakable_kern:n \__xeCJK_punct_breakable_kern:n
6716     \tl_set:Nc \l__xeCJK_off_verb_addon_tl
6717     {
6718         \bool_if:NTF \l__xeCJK_xecglue_bool
6719         { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
6720         { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
6721         \exp_not:n
6722         {
6723             \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
6724             \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
6725             \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_shipout_punct_hskip:n
6726             \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n
6727                 \__xeCJK_shipout_punct_breakable_kern:n
6728             \l__xeCJK_reset_shipout_skip_hook_tl
6729         }
6730     }
6731     \xeCJK_add_to_shipout:n { \l__xeCJK_off_verb_addon_tl }
6732     \keys_set:nn { xeCJK / options } { xCJKecglue = false }
6733 }
6734 \tl_new:N \l__xeCJK_reset_shipout_skip_hook_tl

```

`\xeCJKOffVerbAddOn` `\xeCJKVerbAddOn` 进行了比较大的调整, 应该只在分组环境里使用。为了方便调整间距以利于对齐, 这里只把字符分成了两类, 并且在 CJK 类与边界(空格)之间也插入 `\CJKecglue`。以

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

```

6735 \NewDocumentCommand \xeCJKVerbAddon { }
6736 {
6737   \int_compare:nNf \tex_currentgrouplevel:D = \c_zero_int
6738   {
6739     \bool_if:NF \l__xeCJK_listings_env_bool
6740     {
6741       \dim_compare:nNfTF
6742       { \tex_fontdimen:D 2 ~ \tex_font:D } =
6743       { \tex_fontcharwd:D \tex_font:D \c__xeCJK_mono_letter_int }
6744       {
6745         \__xeCJK_set_verb_exspace:
6746         \__xeCJK_verb_addon:
6747       }
6748       {
6749         \int_if_odd:nTF { \l__xeCJK_verb_case_int }
6750         { \__xeCJK_nobreak_skip_zero: }
6751         { \__xeCJK_nobreak_skip: }
6752       }
6753     }
6754   }
6755 }
6756 \int_const:Nn \c__xeCJK_mono_letter_int { 77 }
6757 \bool_new:N \l__xeCJK_listings_env_bool
6758 \NewDocumentCommand \xeCJKOffVerbAddon { }
6759 { \tl_use:N \l__xeCJK_off_verb_addon_tl }
6760 \tl_new:N \l__xeCJK_off_verb_addon_tl
6761 \cs_new_protected:Npn \__xeCJK_verb_addon:
6762 {
6763   \bool_if:NF \l__xeCJK_verb_addon_bool
6764   { \__xeCJK_verb_addon_action: }
6765   \skip_if_eq:nnTF { \l__xeCJK_verb_exspace_skip } { \c_zero_skip }
6766   {
6767     \xeCJK_cs_clear:N \CJKglue
6768     \xeCJK_cs_clear:N \CJKe glue
6769   }
6770   {
6771     \skip_set_eq:NN \l__xeCJK_ccglue_skip \l__xeCJK_verb_exspace_skip
6772     \skip_set:Nn \l__xeCJK_ecglue_skip { \l__xeCJK_verb_exspace_skip / 2 }
6773     \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue:
6774     \cs_set_eq:NN \CJKe glue \__xeCJK_nobreak_ecglue:
6775   }
6776   \cs_set_eq:NN \xeCJK_check_for_glue: \CJKe glue
6777   \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_CJK_and_Boundary:w
6778 }
6779 \cs_new_protected:Npn \__xeCJK_verb_addon_action:
6780 {
6781   \bool_set_true:N \l__xeCJK_verb_addon_bool
6782   \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
6783   \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
6784   \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
6785   \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
6786   \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
6787   \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
6788   \cs_set_eq:NN \__xeCJK_shipout_CJKe glue: \CJKe glue
6789   \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
6790   \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
6791   \cs_set_protected:Npe \xeCJKOffVerbAddon
6792   {
6793     \__xeCJK_reset_char_class:n { FullLeft }
6794     \__xeCJK_reset_char_class:n { FullRight }
6795     \__xeCJK_reset_char_class:n { HalfLeft }
6796     \__xeCJK_reset_char_class:n { HalfLeft }
6797     \__xeCJK_reset_char_class:n { NormalSpace }
6798     \bool_if:Nf \l__xeCJK_xecglue_bool

```

```

6799         { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
6800         { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
6801     \exp_not:n
6802     {
6803         \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
6804         \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
6805         \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
6806         \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
6807     }
6808 }
6809 \xeCJK_add_to_shipout:n { \xeCJKOffVerbAddon }
6810 \keys_set:nn { xeCJK / options } { xCJKecglue = false }
6811 }
6812 \cs_new_protected:Npn \__xeCJK_verb_CJK_and_Boundary:w
6813 { \xeCJK_class_group_end: \CJKecglue }
6814 \cs_new_protected:Npn \__xeCJK_reset_char_class:n #1
6815 {
6816     \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
6817     \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
6818     { \tex_XeTeXcharclass:D ##1 = \l__xeCJK_tmp_int }
6819 }
6820 \bool_new:N \l__xeCJK_verb_addon_bool
6821 \cs_new_eq:NN \CJKfixedspacing \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。

```

6822 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:
6823 {
6824     \tl_if_exist:cTF { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
6825     {
6826         \skip_set:Nn \l__xeCJK_verb_exspace_skip
6827         { \use:c { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size } }
6828     }
6829     {
6830         \tl_set:Nc \l__xeCJK_current_coord_tl { \CJK@family/\curr@fontshape }
6831         \prop_get:NnTF \g__xeCJK_scale_family_prop
6832         \l__xeCJK_current_coord_tl \l__xeCJK_family_tl
6833         {
6834             \xeCJK_switch_family:o { \l__xeCJK_family_tl }
6835             \skip_zero:N \l__xeCJK_verb_exspace_skip
6836         }
6837         {
6838             \group_begin: \xeCJK_select_font: \exp_args:NNo \group_end:
6839             \__xeCJK_set_verb_exspace:n
6840             { \dim_use:N \tex_fontcharwd:D \tex_font:D "4E00 ~ }
6841         }
6842     }
6843 }
6844 \skip_new:N \l__xeCJK_verb_exspace_skip

```

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

```

6845 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:n #1
6846 {
6847     \skip_set:Nn \l__xeCJK_verb_exspace_skip
6848     { 2 \tex_fontdimen:D 2 ~ \tex_font:D - #1 }
6849     \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
6850     {
6851         \skip_zero:N \l__xeCJK_verb_exspace_skip
6852         \exp_args:Nee \__xeCJK_set_verb_scale:nn
6853         { \dim_to_fp:n { 2 \tex_fontdimen:D 2 ~ \tex_font:D } }
6854         { \dim_to_fp:n {#1} }
6855     }
6856     {
6857         \tl_const:ce { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
6858         { \skip_use:N \l__xeCJK_verb_exspace_skip }
6859     }

```

```
6860 }
```

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

```
6861 \cs_new_protected:Npn \_xeCJK\_set\_verb\_scale:nn #1#2
6862 {
6863   \fp_set:Nn \l__xeCJK\_scale\_factor\_fp { #1 / #2 }
6864   \_xeCJK\_warning:nee { scale-factor }
6865   { \fp_eval:n { trunc ( \l__xeCJK\_scale\_factor\_fp , 4 ) } }
6866   { \fp_eval:n { ceil ( #2 / #1 , 4 ) } }
6867   \xeCJK\_add\_font\_features:Nne \c\_true\_bool
6868   { } { Scale = { \fp\_use:N \l__xeCJK\_scale\_factor\_fp } }
6869   \prop\_gput:Noo \g__xeCJK\_scale\_family\_prop
6870   \l__xeCJK\_current\_coord\_tl \l\_xeCJK\_family\_tl
6871 }
6872 \_xeCJK\_msg\_new:nn { scale-factor }
6873 {
6874   \token\_to\_str:N \xeCJKVerbAddon'~may~not~work~properly.\\
6875   You~may~set~`Scale=#1'~to~CJKfamily~
6876   \_xeCJK\_msg\_family\_map:n { \l\_xeCJK\_family\_tl }',\\
6877   or~set~`Scale=#2'~to~family~
6878   \str\_if\_eq:eeTF \f@family \ttdefault
6879   { \token\_to\_str:N \ttdefault } { \f@family }'.
6880 }
6881 \fp\_new:N \l__xeCJK\_scale\_factor\_fp
6882 \prop\_new:N \g__xeCJK\_scale\_family\_prop
```

\xeCJK_visible_space: 如果文档不使用 EU1 作为默认字体编码,那么默认的打字机字体族很可能是传统的 TeX 字体,
 \@setupverbvisiblespace 这时可视空格按照 OT1 编码传统一般就是字体中的\char32。

```
6883 \cs_new_protected:Npn \xeCJK\_setup\_visible\_space:
6884 {
6885   \xeCJK\_make\_boundary:
6886   \xeCJK\_glyph\_if\_exist:NTF { ~~~~2423 }
6887   { \tl\_set:Nn \l__xeCJK\_visible\_space\_tl { ~~~~2423 } }
6888   {
6889     \int\_compare:nNnTF { \tex\_XeTeXfonttype:D \tex\_font:D } = \c\_zero\_int
6890     {
6891       \tl\_set:Nn \l__xeCJK\_visible\_space\_tl
6892       {
6893         \str\_if\_eq:eeTF { \f@family } { \ttdefault }
6894         { \c\_catcode\_other\_space\_tl }
6895         { \exp\_not:N \textvisiblespace }
6896       }
6897     }
6898     { \_xeCJK\_visible\_space\_fallback: }
6899   }
6900   \cs\_set\_eq:NN \@xobeysp \l__xeCJK\_visible\_space\_tl
6901 }
6902 \tl\_new:N \l__xeCJK\_visible\_space\_tl
6903 \cs\_set\_eq:NN \@setupverbvisiblespace \xeCJK\_setup\_visible\_space:
```

_xeCJK_visible_space_fallback: 我们使用 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,这里需要做一些调整,以保证使用后备可视空格符号时,也能保证对齐。

```
6904 \cs_new_protected:Npn \_xeCJK\_visible\_space\_fallback:
6905 {
6906   \exp\_args:Nc \_xeCJK\_visible\_space\_fallback\_auxi:N
6907   { xeCJK/space/\curr@fontshape/\f@size }
6908 }
6909 \cs_new_protected:Npn \_xeCJK\_visible\_space\_fallback\_auxi:N #1
6910 {
6911   \cs\_if\_exist:NF #1
6912   { \_xeCJK\_visible\_space\_fallback\_auxii:N #1 }
6913   \tl\_set:Nn \l__xeCJK\_visible\_space\_tl {#1}
6914 }
```

_xeCJK_visible_space_fallback_auxii:N

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。

```

6915 \cs_new_protected:Npn \_xeCJK\_visible\_space\_fallback\_auxii:N #1
6916 {
6917   \group_begin:
6918     \exp_args:No \_xeCJK\_set\_visible\_space\_size:n
6919     { \dim_use:N \tex_fontdimen:D 2 ~ \tex_font:D }
6920     \cs_new_protected:Npe #1
6921     { \group_begin: \tex_the:D \tex_font:D ~~~~2423 \group_end: }
6922   \group_end:
6923 }
6924 \cs_new_protected:Npn \_xeCJK\_set\_visible\_space\_size:n #1
6925 {
6926   \fontencoding { \UnicodeEncodingName }
6927   \tl_set:Nn \f@family { lmtt }
6928   \selectfont
6929   \dim_compare:nNnF {#1} = { \tex_fontdimen:D 2 ~ \tex_font:D }
6930   {
6931     \fontsize
6932     {
6933       \dim_eval:n
6934       {
6935         \f@size pt *
6936         \dim_ratio:nn {#1} { \tex_fontdimen:D 2 ~ \tex_font:D }
6937       }
6938     }
6939     { \f@baselineskip }
6940   \selectfont
6941 }
6942 }

```

5.16 xeCJK 其他选项

LocalConfig 声明载入本地配置文件的选项。

```

6943 \keys_define:nn { xeCJK / options }
6944 {
6945   LocalConfig .choice: ,
6946   LocalConfig / false .code:n =
6947   { \bool_gset_false:N \g__xeCJK_config_bool } ,
6948   LocalConfig / true .code:n =
6949   {
6950     \bool_gset_true:N \g__xeCJK_config_bool
6951     \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
6952   } ,
6953   LocalConfig / unknown .code:n =
6954   {
6955     \bool_gset_true:N \g__xeCJK_config_bool
6956     \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
6957   } ,
6958   LocalConfig .default:n = { true }
6959 }
6960 \tl_new:N \g__xeCJK_config_name_tl
6961 \bool_new:N \g__xeCJK_config_bool

```

CJKnumber **CJKnumber** 和 **indentfirst** 是过时选项。

```

6962 \keys_define:nn { xeCJK / options }
6963 {
6964   CJKnumber .code:n =
6965   { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { CJKnumb } } ,
6966   indentfirst .code:n =
6967   { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { indentfirst } } ,
6968   normalindentfirst .code:n =
6969   { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { } }
6970 }

```

```

6971 \__xeCJK_msg_new:nn { option-deprecated }
6972 {
6973   The~`#1'~option~is~deprecated.\\
6974   \tl_if_empty:nF {#2}
6975   { You~may~load~the~package~`#2'~after~xeCJK~to~use~its~function.\\ }
6976 }

```

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

```

6977 \keys_define:nn { xeCJK / options }
6978 {
6979   quiet .code:n =
6980   {
6981     \msg_redirect_module:nnn { xeCJK } { warning } { info }
6982     \msg_redirect_module:nnn { xeCJK } { info } { none }
6983     \xeCJK_if_package_loaded:nF { fontspec }
6984     { \PassOptionsToPackage { quiet } { fontspec } }
6985   } ,
6986   silent .code:n =
6987   {
6988     \msg_redirect_module:nnn { xeCJK } { warning } { none }
6989     \msg_redirect_module:nnn { xeCJK } { info } { none }
6990     \xeCJK_if_package_loaded:nF { fontspec }
6991     { \PassOptionsToPackage { silent } { fontspec } }
6992   } ,
6993   unknown .code:n =
6994   {
6995     \xeCJK_if_package_loaded:nTF { fontspec }
6996     { \__xeCJK_error:ne { key-unknown } { \l_keys_key_str } }
6997     { \PassOptionsToPackage { \l_keys_key_str } { fontspec } }
6998   }
6999 }
7000 \__xeCJK_msg_new:nn { key-unknown }
7001 {
7002   Sorry,~but~xeCJK/options~does~not~have~a~key~called~`#1'.\\
7003   The~key~`#1'~is~being~ignored.
7004 }

```

5.17 xeCJK 初始化设置

```

\CJKsymbol 7005 \cs_new_eq:NN \CJKsymbol \use:n
\CJKpunctsymbol 7006 \cs_new_eq:NN \CJKpunctsymbol \use:n

```

xeCJK 宏包的初始化设置。

```

7007 \keys_set:nn { xeCJK / options }
7008 {
7009   CJKglue = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
7010   CJKecglue = { ~ } ,
7011   xCJKecglue = false ,
7012   CheckSingle = false ,
7013   PlainEquation = false ,
7014   CheckFullRight = false ,
7015   CJKspace = false ,
7016   CJKmath = false ,
7017   xeCJKactive = true ,
7018   LocalConfig = true ,
7019   LoadFandol = true ,
7020   RubberPunctSkip = true ,
7021   Verb = env ,
7022   EmboldenFactor = 4 ,
7023   SlantFactor = 0.167 ,
7024   PunctStyle = quanjiao ,
7025   NewLineCS = { \par \[ } ,
7026   EnvCS = { \begin \end } ,
7027   WidowPenalty = { 10 000 } ,

```

```

7028 NoBreakCS      = { \footnote \footnotemark \nobreak } ,
7029 KaiMingPunct    = { ~~~~3002 ~~~~ff0e ~~~~ff1f ~~~~ff01 } ,
7030 LongPunct       = { ~~~~2014 ~~~~2e3a ~~~~2025 ~~~~2026 } ,
7031 NoBreakLongPunct = { ~~~~2025 ~~~~2026 } ,
7032 MiddlePunct     = { ~~~~2013 ~~~~2014 ~~~~2e3a ~~~~2027 ~~~~00b7 ~~~~30fb ~~~~ff65 } ,
7033 AllowBreakBetweenPuncts = false
7034 }
7035 \defaultCJKfontfeatures { Script = CJK }

```

半字线连接号¹⁶应为半角宽度。

```
7036 \xeCJKsetwidth { ~~~~2013 } { 0.5 em }
```

执行宏包选项,并载入 fontspec 宏包。

```

7037 \cs_if_exist:NTF \ProcessKeyOptions
7038 { \ProcessKeyOptions [ xeCJK / options ] }
7039 {
7040   \RequirePackage { l3keys2e }
7041   \ProcessKeysOptions { xeCJK / options }
7042 }
7043 \RequirePackage { fontspec } [ 2020/02/03 ]

```

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

```
7044 \tl_const:Nx \c__xeCJK_encoding_tl { \g_fontspec_encoding_tl }
```

对不能通过 `\xeCJKsetup` 设置的选项给出警告。

```

7045 \keys_define:nn { xeCJK / options }
7046 {
7047   LocalConfig .code:n =
7048     { \__xeCJK_warning:ne { option-invalid } { \l_keys_key_str } }
7049 }
7050 \__xeCJK_msg_new:nn { option-invalid }
7051 {
7052   The~`#1'~option~can~only~be~set~in~the~optional~argument~to~the\\
7053   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
7054   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
7055 }

```

```

\CJKrmdefault 7056 \tl_if_exist:NF \CJKrmdefault
\CJKsfdefault 7057 { \tl_new:N \CJKrmdefault \tl_gset:Nn \CJKrmdefault { rm } }
\CJKttdefault 7058 \tl_if_exist:NF \CJKsfdefault
\CJKfamilydefault 7059 { \tl_new:N \CJKsfdefault \tl_gset:Nn \CJKsfdefault { sf } }
7060 \tl_if_exist:NF \CJKttdefault
7061 { \tl_new:N \CJKttdefault \tl_gset:Nn \CJKttdefault { tt } }
7062 \tl_new:N \l__xeCJK_family_default_init_tl
7063 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
7064 \tl_set:Nx \l__xeCJK_family_default_init_tl
7065 {
7066   \exp_not:N \__xeCJK_family_default_wrap:n
7067   {
7068     \tl_if_exist:NTF \CJKfamilydefault
7069     { \exp_not:o \CJKfamilydefault }
7070     { \exp_not:N \CJKrmdefault }
7071   }
7072 }
7073 \tl_if_exist:NF \CJKfamilydefault
7074 { \tl_new:N \CJKfamilydefault }
7075 \tl_gset_eq:NN \CJKfamilydefault \l__xeCJK_family_default_init_tl

```

`\xeCJKsetup` 在导言区或文档中设置 xeCJK 的接口。

```

7076 \NewDocumentCommand \xeCJKsetup { +m }
7077 {
7078   \keys_set:nn { xeCJK / options } {#1}
7079   \tex_ignorespaces:D
7080 }

```

¹⁶见《夹用英文的中文文本的标点符号用法(草案)》5.13 节。

```

\XeCJKsetemboldenfactor 7081 \NewDocumentCommand \XeCJKsetemboldenfactor { m }
\XeCJKsetslantfactor 7082 { \XeCJKsetup { EmboldenFactor = {#1} } }
7083 \NewDocumentCommand \XeCJKsetslantfactor { m }
7084 { \XeCJKsetup { SlantFactor = {#1} } }

\punctstyle 7085 \NewDocumentCommand \punctstyle { m } { \XeCJKsetup { PunctStyle = {#1} } }
\XeCJKplainchr 7086 \NewDocumentCommand \XeCJKplainchr { } { \XeCJKsetup { PunctStyle = plain } }

\CJKsetecglue 7087 \NewDocumentCommand \CJKsetecglue { m } { \XeCJKsetup { CJKecglue = {#1} } }
7088 \cs_new_eq:NN \XeCJKsetecglue \CJKsetecglue

\CJKspace 7089 \NewDocumentCommand \CJKspace { } { \XeCJKsetup { CJKspace = true } }
\CJKnospace 7090 \NewDocumentCommand \CJKnospace { } { \XeCJKsetup { CJKspace = false } }

\XeCJKallowbreakbetweenpuncts 7091 \NewDocumentCommand \XeCJKallowbreakbetweenpuncts { }
\XeCJKnobreakbetweenpuncts 7092 { \XeCJKsetup { AllowBreakBetweenPuncts = true } }
7093 \NewDocumentCommand \XeCJKnobreakbetweenpuncts { }
7094 { \XeCJKsetup { AllowBreakBetweenPuncts = false } }

\XeCJKenablefallback 7095 \NewDocumentCommand \XeCJKenablefallback { }
\XeCJKdisablefallback 7096 { \XeCJKsetup { AutoFallBack = true } }
7097 \NewDocumentCommand \XeCJKdisablefallback { }
7098 { \XeCJKsetup { AutoFallBack = false } }

\XeCJKsetcharclass 7099 \__XeCJK_msg_new:nn { XeCJKsetcharclass-deprecated }
7100 {
7101   \token_to_str:N \XeCJKsetcharclass\ is~deprecated~and~
7102   does~not~update~CJKmath~char~lists.\\
7103   Use~\token_to_str:N \XeCJKDeclareCharClass\ instead.
7104 }
7105 \NewDocumentCommand \XeCJKsetcharclass { m m m }
7106 {
7107   \__XeCJK_error:n { XeCJKsetcharclass-deprecated }
7108   \XeCJK_set_char_class:nnn {#1} {#2} {#3}
7109   \XeCJKResetPunctClass
7110 }

```

5.18 兼容性修补

`\XeCJK@update@fam` 使通过 `\urlstyle` 或者 `\UrlFont` 设置的路径中使用的 CJK 字体生效。使用 `\everymath` 钩子中数学模式中重定义 CJK 数学字体, 以确保我们的设置在 `\check@mathfonts` 之后生效, 不会被它覆盖。更合理的方式是定义一个新的 `\mathversion` 来切换。

```

7111 \cs_new_protected:Npn \XeCJK@update@fam
7112 {
7113   \addto@hook \everymath
7114   {
7115     \__XeCJK_update_main_fam:
7116     \__XeCJK_update_block_fam:
7117   }
7118 }
7119 \cs_new_protected:Npn \__XeCJK_update_main_fam:
7120 {
7121   \group_begin:
7122   \XeCJK_select_font:
7123   \exp_last_unbraced:NNNo \group_end:
7124   \tex_textfont:D \c_XeCJK_math_fam_int \tex_the:D \tex_font:D
7125 }
7126 \cs_new_protected:Npn \__XeCJK_update_block_fam:
7127 {
7128   \prop_if_empty:NF \g__XeCJK_block_fam_prop
7129   {
7130     \prop_map_function:NN
7131     \g__XeCJK_block_fam_prop
7132     \__XeCJK_update_block_fam:nn
7133   }
7134 }

```

```

7135 \cs_new_protected:Npn \__xeCJK_update_block_fam:nn #1#2
7136 {
7137   \int_set:Nn \l__xeCJK_fam_int {#2}
7138   \group_begin:
7139     \xeCJK_select_font:n {#1}
7140     \exp_last_unbraced:NNNo \group_end:
7141     \tex_textfont:D \l__xeCJK_fam_int \tex_the:D \tex_font:D
7142   }
7143 \__xeCJK_after_end_preamble:n
7144 {
7145   \bool_lazy_and:nnT
7146     { \g__xeCJK_math_bool }
7147     { \cs_if_exist_p:N \Url@MathSetup }
7148     { \tl_put_right:Nn \Url@MathSetup { \xeCJK@update@fam } }
7149 }

```

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

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

`\math`

`\endmath`

`\ensuremath`

`\__xeCJK_math_robust:N`

`\__xeCJK_boundary_math_begin:`

`\__xeCJK_boundary_math_begin:n`

`\__xeCJK_boundary_register_text_command:N`

`\__xeCJK_boundary_register_text_command:NN`

这个定义最开始的 `\relax` 是为了防止 `\(` 出现在表格单元格的开始位置时, 模式判断不正确 (因为 $\text{\TeX}$ 会先看单元格中第一个不可展的非空格记号是否是 `\omit` 或 `\noalign`)。但是它会 造成一个边界, 使 *xeCJK* 不能看到 `\relax` 后面出现的 `$`, 从而不能加入间距¹⁷。使用 $\epsilon\text{-TeX}$ 的 `\protected` 来定义它, 可以不需要 `\relax`, 或者将 `\relax` 改成 `\scan_align_safe_stop:`, 都可以避免这些情况。同时 `fixltx2e` 中还使用了 `\MakeRobust\()`, 我们需要小心处理。另外 *ulem* 也定义了一个 `\MakeRobust`, 如果它被放在 `fixltx2e` 之前载入, 那么 `fixltx2e` 的定义就会 失效 (因为 `fixltx2e` 使用 `\providecommand*` 来定义 `\MakeRobust`)。但是 *ulem* 的定义并不完 全正确, 没有考虑 $\text{\TeX}$ 不会略去控制符号后面的空格的情况。

```

7150 \cs_new_protected:Npn \__xeCJK_math_robust:N #1
7151 {
7152   \group_begin: \exp_args:NcNc \group_end:
7153     { \__xeCJK_math_robust_aux:NN } #1 { \cs_to_str:N #1 ~ }
7154 }
7155 \cs_new_protected:Npn \__xeCJK_math_robust_aux:NN #1#2
7156 {
7157   \exp_args:Ne \str_case:nnTF { \cs_replacement_spec:N #1 }
7158   {
7159     { \x@protect #1 \protect #2 } { }
7160     { \protect #2 } { }
7161   }
7162   { \__xeCJK_math_robust:NN #1#2 }
7163   { \__xeCJK_math_robust:NN #1#1 }
7164 }
7165 \cs_new_protected:Npn \__xeCJK_math_robust:NN #1#2
7166 {
7167   \str_if_eq:eeTF { \cs_argument_spec:N #2 } { }
7168   {
7169     \exp_args:No \tl_if_head_eq_meaning:nNTF {#2} \scan_stop:
7170     {
7171       \cs_gset_protected:Npe #1
7172       { \tl_tail:N #2 }
7173     }
7174     {
7175       \cs_if_eq:NNTF #1 \ensuremath
7176       {
7177         \cs_gset_protected:Npe #1
7178         { \exp_not:o {#2} }
7179       }
7180       {
7181         \__xeCJK_warning:nee { robust-failure }
7182         { \token_to_str:N #1 } { \token_to_meaning:N #2 }
7183       }

```

¹⁷<http://tex.stackexchange.com/q/124773>

```

7184     }
7185   }
7186   {
7187     \__xeCJK_warning:née { robust-failure }
7188     { \token_to_str:N #1 } { \token_to_meaning:N #2 }
7189   }
7190 }
7191 \__xeCJK_msg_new:nnn { robust-failure }
7192 { xeCJK~can~not~make~`#1'~robust. }
7193 {
7194   The~current~meaning~of~`#1'~is:\\
7195   \iow_indent:n {#2}
7196 }
7197 \cs_if_eq:NNTF \(\ \math
7198 {
7199   \__xeCJK_math_robust:N \(\
7200   \cs_set_eq:NN \math \(\
7201 }
7202 {
7203   \__xeCJK_math_robust:N \(\
7204   \__xeCJK_math_robust:N \math
7205 }
7206 \cs_if_eq:NNTF \) \endmath
7207 {
7208   \__xeCJK_math_robust:N \)
7209   \cs_set_eq:NN \endmath \)
7210 }
7211 {
7212   \__xeCJK_math_robust:N \)
7213   \__xeCJK_math_robust:N \endmath
7214 }
7215 \__xeCJK_math_robust:N \ensuremath
7216 % stream 的左侧间距必须排在 math-on 节点之前。三个显式公式入口只处理
7217 % 当前列表的边界，再调用已经完成 robust 处理的原命令；它们不直接向所有
7218 % 外层 capture 报告，否则原语 \tn{setbox} 中不可见的测量公式也会被当成
7219 % 外层输出。已经注册的命令分别检查可见正文的开头和结尾：开头是 |$|、
7220 % \tn{ } 或 \tn{ensuremath} 时，在 math-on 前报告首类别；结尾是对应公式
7221 % 时，把 |math| 记为当前 capture 的确定末类别。 \tn{ensuremath} 已在数学
7222 % 模式中时不重复处理当前列表。
7223 \clist_map_inline:nn { ( , math , ensuremath }
7224 { \__xeCJK_boundary_reserve:n {#1} }
7225 \cs_if_eq:NNTF \(\ \math
7226 {
7227   \cs_new_eq:NN \__xeCJK_orig_inline_math_begin: \(\
7228   \cs_gset_protected:Npn \(\
7229     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_inline_math_begin: }
7230   \cs_gset_eq:NN \math \(\
7231 }
7232 {
7233   \cs_new_eq:NN \__xeCJK_orig_inline_math_begin: \(\
7234   \cs_gset_protected:Npn \(\
7235     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_inline_math_begin: }
7236   \cs_new_eq:NN \__xeCJK_orig_math_begin: \math
7237   \cs_gset_protected:Npn \math
7238     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_math_begin: }
7239 }
7240 \cs_new_eq:NN \__xeCJK_orig_ensuremath: \ensuremath
7241 \cs_gset_protected:Npn \ensuremath
7242 {
7243   \mode_if_math:TF
7244     { \__xeCJK_orig_ensuremath: }
7245     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_ensuremath: }
7246 }
7247 % LaTeX 标准文本字体命令的 robust 内部实现均为单参数。这里保存该内部实现，
7248 % 用 stream 包住公式位于正文任一边缘的调用；开头公式在 math-on 前报告，
7249 % 结尾公式在正文参数实际排完时确认。普通正文仍由 interchar 转换报告实际
7250 % 首尾类别。

```

```

7251 \cs_new_protected:Npn \__xeCJK_boundary_register_text_command:N #1
7252 {
7253   \group_begin: \exp_args:NcNc \group_end:
7254   { \__xeCJK_boundary_register_text_command:NN }
7255   #1 { \cs_to_str:N #1 ~ }
7256 }
7257 \cs_new_protected:Npn \__xeCJK_boundary_register_text_command:NN #1#2
7258 {
7259   \__xeCJK_boundary_reserve:N #1
7260   \__xeCJK_boundary_reserve:N #2
7261   \exp_args:Nc \cs_new_eq:NN
7262   { \__xeCJK_boundary_orig_\cs_to_str:N #1 :n } #2
7263   \cs_gset_protected:Npn #2 ##1
7264   {
7265     \__xeCJK_boundary_if_math_edge:nTF {##1}
7266     {
7267       \__xeCJK_boundary_inline_stream_begin:n { auto }
7268       \__xeCJK_boundary_math_begin:n {##1}
7269       \use:c { \__xeCJK_boundary_orig_\cs_to_str:N #1 :n }
7270       { ##1 \__xeCJK_boundary_math_end:n {##1} }
7271       \__xeCJK_boundary_inline_stream_end:n { auto }
7272     }
7273     { \use:c { \__xeCJK_boundary_orig_\cs_to_str:N #1 :n } {##1} }
7274   }
7275 }
7276 \clist_map_inline:nn
7277 {
7278   \textrm , \textsf , \texttt , \textnormal , \textbf , \textmd ,
7279   \textit , \textsl , \textsc , \textup , \emph
7280 }
7281 { \__xeCJK_boundary_register_text_command:N #1 }

```

`\fontfamily` 使用 L^AT_EX 2_ε 2020/10/01 提供的 NFSS 钩子, 使主要 CJK 字体族能随西文主要字体更新。
`\xeCJK@family`

```

7282 \cs_set_eq:NN \xeCJK@family \xeCJK_switch_family:e
7283 \ctex_gadd_ltxhook:nn { rmfamily } { \xeCJK@family { \CJKrmdefault } }
7284 \ctex_gadd_ltxhook:nn { sffamily } { \xeCJK@family { \CJKsfdefault } }
7285 \ctex_gadd_ltxhook:nn { ttfamily } { \xeCJK@family { \CJKttdefault } }
7286 \ctex_gadd_ltxhook:nn { normalfont } { \xeCJK@family { \CJKfamilydefault } }
7287 <@@=>

```

`\xeCJK@fix@penalty` L^AT_EX 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 `\CJKecglue` 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。

```

7288 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
7289 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
7290 \tl_replace_once:Nnn \sw@slant { \fix@penalty } { \xeCJK@fix@penalty }

```

`\xeCJK@italiccorr` 修复倾斜校正, 并处理汉字后面的空格。

```

7291 \cs_new_protected:Npn \xeCJK@italiccorr
7292 {
7293   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7294   { \xeCJK_italic_correction: }
7295   { \@@italiccorr }
7296 }
7297 <@@=\xeCJK>

```

`\xeCJK_italic_correction:` 修复倾斜校正, 并处理汉字后面的空格。

```

7298 \cs_new_protected:Npn \xeCJK_italic_correction:
7299 { \__xeCJK_if_last_kern:T { \__xeCJK_italic_correction: } }
7300 \cs_new_protected:Npn \__xeCJK_italic_correction:
7301 {
7302   \dim_case:nnF { \tex_lastkern:D }
7303   {
7304     { \__xeCJK_node:n { default } }
7305     {
7306       \xeCJK_remove_node: \tex_italiccorrection:D
7307       \xeCJK_make_node:n { default }
7308     }
7309     { \__xeCJK_node:n { CJK } }
7310     {
7311       \xeCJK_remove_node: \tex_italiccorrection:D
7312       \xeCJK_make_node:n { CJK }
7313       \__xeCJK_italic_correction_aux:
7314     }
7315     { \__xeCJK_node:n { CJK-space } }
7316     {
7317       \xeCJK_remove_node: \tex_italiccorrection:D
7318       \xeCJK_make_node:n { CJK-space }
7319       \__xeCJK_italic_correction_aux:
7320     }
7321   }
7322   { \tex_italiccorrection:D }
7323 }

```

`\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:`)。

```

7324 \cs_new_protected:Npn \__xeCJK_italic_correction_aux:
7325 {
7326   \exp_after:wN \exp_after:wN \exp_after:wN
7327   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7328   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7329   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7330   \xeCJK_ignore_spaces:w
7331 }

```

`\g__xeCJK_xetex_allocator_int` $\text{\LaTeX}2_{\epsilon}$ 2015/01/01 接管了 `\newXeTeXintercharclass`。

```

7332 \cs_new_eq:NN \g__xeCJK_xetex_allocator_int \xe@alloc@intercharclass

```

`\__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。

```

7333 \__xeCJK_after_end_preamble:n
7334 {
7335   \int_compare:nNnF
7336   { \c__xeCJK_class_begin_int + \seq_count:N \g__xeCJK_new_class_seq } =
7337   { \g__xeCJK_xetex_allocator_int }
7338   {
7339     \int_step_inline:nnn
7340     { \c__xeCJK_class_begin_int + 1 }
7341     { \g__xeCJK_xetex_allocator_int }
7342     {
7343       \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
7344       { \__xeCJK_set_others_toks:n {#1} }
7345     }
7346   }
7347 }
7348 \cs_new_protected:Npn \__xeCJK_set_others_toks:n #1
7349 {

```

```

7350 \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
7351 \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
7352 {
7353   \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
7354   \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
7355   \xeCJK_app_inter_class_toks:nne {##1} { Others }
7356   { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
7357   \xeCJK_pre_inter_class_toks:nne { Others } {##1}
7358   { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
7359   \tl_if_blank:eT
7360   { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
7361   {
7362     \xeCJK_copy_inter_class_toks:nnnn
7363     { Others } { Boundary } { Default } { Boundary }
7364   }
7365   \tl_if_blank:eT
7366   { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
7367   {
7368     \xeCJK_copy_inter_class_toks:nnnn
7369     { Boundary } { Others } { Boundary } { Default }
7370   }
7371 }
7372 }

```

用于保护下面歧义宽度标点的分组。

```

7373 \cs_new_protected:Npn \__xeCJK_inactive_group_begin:
7374 { \group_begin: \makeXeCJKinactive }
7375 \cs_new_eq:NN \__xeCJK_inactive_group_end: \group_end:

```

单独处理宽度有分歧的几个标点：包括省略号、破折号、间隔号、引号等中西文混用的符号，保证其命令形式输出的是西文字体。如果 xunicode 宏包被载入，则通过 xunicode-addon 处理。

```

7376 \prop_const_from_keyval:Nn \c__xeCJK_ambiguous_char_prop
7377 {
7378   "00B7 = \textperiodcentered \textcentereddot \textcdot ,
7379   "2013 = \textendash ,
7380   "2014 = \textemdash ,
7381   "2018 = \textquoteleft \textgrq ,
7382   "2019 = \textquoteright ,
7383   "201C = \textquotedblleft \textgrqq ,
7384   "201D = \textquotedblright ,
7385   "2025 = \texthdofor ,
7386   "2026 = \textellipsis ,
7387   "2027 = \texthyphenationpoint ,
7388   "2E3A = \texttwoemdash
7389 }
7390 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_text_command: }
7391 \cs_new_protected:Npn \__xeCJK_patch_text_command:
7392 {
7393   \xeCJK_if_package_loaded:nTF { xunicode }
7394   { \__xeCJK_patch_xunicode_ambiguous_char: }
7395   {
7396     \exp_args:Ne \__xeCJK_patch_tuenc_ambiguous_char:n
7397     { \UnicodeEncodingName }
7398     \__xeCJK_patch_tuenc_accent:
7399     \__xeCJK_patch_tuenc_composite:
7400   }
7401 }
7402 \cs_new_protected:Npn \__xeCJK_patch_xunicode_ambiguous_char:
7403 {
7404   \RequirePackage { xunicode-addon }
7405   \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
7406   {
7407     \tl_map_inline:nn { ##2 }
7408     {
7409       \xunadd_set_begin_hook:nn { ####1 }
7410       { \__xeCJK_inactive_group_begin: }

```

```

7411         \xunadd_set_end_hook:nn { #####1 }
7412         { \__xeCJK_inactive_group_end: }
7413     }
7414 }
7415 \xunadd_append_begin_hook:n { \xeCJK_make_boundary: }
7416 }
7417 \cs_new_protected:Npn \__xeCJK_patch_tuenc_ambiguous_char:n #1
7418 {
7419     \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
7420     {
7421         \tl_map_inline:nn { ##2 }
7422         {
7423             \cs_if_exist:NF #####1
7424             { \DeclareTextSymbol #####1 {#1} { ##1 } }
7425             \__xeCJK_patch_ambiguous_char:nN {#1} #####1
7426         }
7427     }
7428 }
7429 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nN #1#2
7430 {
7431     \exp_args:Ne \__xeCJK_patch_ambiguous_char:nn
7432     { #1 \token_to_str:N #2 }
7433     { #1 - #2 }
7434 }
7435 \cs_new_protected:Npe \__xeCJK_patch_ambiguous_char:nNn #1#2#3
7436 {
7437     \exp_not:N \exp_args:Ne
7438     \exp_not:N \__xeCJK_patch_ambiguous_char:nn
7439     {
7440         \c_backslash_str #1
7441         \exp_not:N \token_to_str:N #2 -
7442         \exp_not:N \token_to_str:N #3
7443     }
7444     { #1 - #2#3 }
7445 }
7446 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nn #1#2
7447 {
7448     \cs_if_free:cF {#1}
7449     { \exp_args:Nc \__xeCJK_patch_ambiguous_char:Nn {#1} {#2} }
7450 }
7451 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:Nn #1#2
7452 {
7453     \token_if_chardef:NTF #1
7454     {
7455         \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
7456         { \int_eval:n {#1} }
7457         \cs_set_protected:Npe #1
7458         { \__xeCJK_ambiguous_char:n { \tex_Uchar:D #1 } }
7459     }
7460     {
7461         \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
7462         { \int_eval:n { \exp_after:wN ` #1 } }
7463         \cs_set_protected:Npe #1
7464         { \__xeCJK_ambiguous_char:n { \exp_not:o {#1} } }
7465     }
7466 }
7467 \cs_new_protected:Npn \__xeCJK_ambiguous_char:n #1
7468 {
7469     \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7470     { \__xeCJK_inactive_group_begin: #1 \__xeCJK_inactive_group_end: }
7471     {#1}
7472 }
7473 \prop_new:N \g__xeCJK_ambiguous_slot_prop

```

__xeCJK_patch_tuenc_composite: \DeclareUnicodeComposite 具有检查字符是否存在的功能, 当符号命令紧跟在 CJK 字符类

后面时, 需要使字体回到西文状态

```
7474 \cs_new_protected:Npn \xeCJK_text_composite_patch:
7475 {
7476   \str_if_eq:eeT { \f@encoding } { \UnicodeEncodingName }
7477   { \xeCJK_make_boundary: }
7478 }
```

注意 \xeCJK_text_composite_patch: 可能会结束分组, 从而导致 ##1 没有定义时是 \undefined 而不是 \relax, 所以不能将它与 \relax 作比较。

```
7479 \cs_new_protected:Npn \__xeCJK_patch_tuenc_composite:
7480 {
7481   \cs_set_nopar:Npn \@text@composite@x
7482   {
7483     \xeCJK_text_composite_patch:
7484     \cs_if_exist_use:NF
7485   }
7486 }
```

__xeCJK_patch_tuenc_accent: \add@unicode@accent 定义最后用于截断数字展开的 \relax 会造成边界, 可能会影响组合标记。

```
7487 \group_begin:
7488 \char_set_catcode_other:n { "A0 }
7489 \cs_new_protected:Npn \__xeCJK_patch_tuenc_accent:
7490 {
7491   \cs_set_protected_nopar:Npn \add@unicode@accent ##1 ##2
7492   {
7493     \tl_if_blank:nTF { ##2 } { ^^a0 } { ##2 }
7494     \tex_Uchar:D \tex_numexpr:D ##1 \scan_stop:
7495   }
7496 }
7497 \group_end:
```

__xeCJK_patch_middle_dot: 常被用作中文间隔号的 U+00B7 与 T1 等旧字体编码下定义的符号命令冲突。在 encguide.pdf 的编码符号表中, 如下定义有冲突。

```
\DeclareTextComposite{\r}{T1}{u}{183}
\DeclareTextSymbol{\cyrchvcrs}{T2A}{183}
\DeclareTextSymbol{\cyrchldsc}{T2B}{183}
\DeclareTextSymbol{\cyrabhha}{T2C}{183}
\DeclareTextSymbol\textvibyy{T3}{183}
\DeclareTextComposite{\B}{T4}{t}{183}
\DeclareTextComposite{\`}{T5}{\ecircumflex}{183}
\DeclareTextDoubleComposite{\`}{T5}{\^}{e}{183}
\DeclareTextSymbol{\textperiodcentered}{TS1}{183}
\DeclareTextSymbol{\cyrchldsc}{X2}{183}
\DeclareTextSymbol{\textperiodcentered}{LY1}{183}
```

LGR 编码的符号表有 183 号字符, 但在 lgrenc.def 中未找到相应的符号命令。

```
7498 \prop_const_from_keyval:Nn \c__xeCJK_middle_dot_prop
7499 {
7500   T2A = \cyrchvcrs ,
7501   T2B = \cyrchldsc ,
7502   T2C = \cyrabhha ,
7503   X2  = \cyrchldsc ,
7504   TS1 = \textperiodcentered ,
7505   LY1 = \textperiodcentered ,
7506   T1  = \r u ,
7507   T4  = \B t ,
7508   T5  = \` \ecircumflex
7509 }
7510 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_middle_dot: }
7511 \cs_new_protected:Npn \__xeCJK_patch_middle_dot:
7512 {
7513   \prop_map_inline:Nn \c__xeCJK_middle_dot_prop
7514   { \__xeCJK_patch_middle_dot:nw { ##1 } ##2 \q_stop }
```

```

7515 \__xeCJK_patch_ambiguous_char:nNn { T5 } \` { \^ - e }
7516 }
7517 \cs_new_protected:Npn \__xeCJK_patch_middle_dot:nw #1#2#3 \q_stop
7518 {
7519 \tl_if_empty:nTF {#3}
7520 { \__xeCJK_patch_ambiguous_char:nN {#1} #2 }
7521 { \__xeCJK_patch_ambiguous_char:nNn {#1} #2 {#3} }
7522 }

```

pifont 宏包的符号\ding{183} 也有冲突。

```

7523 \__xeCJK_package_hook:nn { pifont }
7524 {
7525 \RenewDocumentCommand \Pifont { m }
7526 { \mode_leave_vertical: \makeXeCJKinactive \usefont { U } {#1} { m } { n } }
7527 }

```

\xeCJKchar \char 是 T_EX 原语, 用户使用它直接通过字符编码从当前字体取字形, 不经过 NFSS 层。一些宏包(如 m_Tpro2)在数学模式中用 \char 取非 CJK 字形时, 会被 xeCJK 的 interchar 机制拦截并切换到中文字体, 导致字形变成中文标点(如间隔号, #407)。

参照 LuaT_EX-j_a 提供 \ltjalchar /\ltjjachar 新命令的思路, xeCJK 提供 \xeCJKchar 命令, 在输出字符前临时关闭 XeTeXinterchartokenstate, 保证输出的字符来自当前字体而不被 interchar 拦截。与 LuaT_EX-j_a 不同的是, XeTeX 的 interchar 机制工作在 token 层面, 没有节点级回调, 因此无法自动区分 \char 和直接字符输入。 \xeCJKchar 供用户和第三方宏包显式使用, \char 本身不做任何修改。

用法与 \char 完全相同: \xeCJKchar"00B7 或 \xeCJKchar183。用户指南和手动补丁方法参见“已知问题和兼容性”一节。

```

7528 \cs_new_protected:Npn \xeCJK_default_char:w
7529 {
7530 \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7531 {
7532 \c_group_begin_token
7533 \tex_afterassignment:D \__xeCJK_default_char_aux:
7534 \l__xeCJK_default_char_int =
7535 }
7536 { \tex_char:D }
7537 }
7538 \cs_new_protected:Npn \__xeCJK_default_char_aux:
7539 {
7540 \tex_XeTeXinterchartokenstate:D = \c_zero_int
7541 \tex_char:D \l__xeCJK_default_char_int
7542 \c_group_end_token
7543 }
7544 \int_new:N \l__xeCJK_default_char_int
7545 \cs_new_eq:NN \xeCJKchar \xeCJK_default_char:w

```

m_Tpro2 的 \overbrace 和 \underbrace 通过 \char 访问 mt2exe 等字体的特殊位置, 其中部分位置(如 183 = U+00B7)与 xeCJK 的 interchar 字符分类冲突。由于这两个命令只用在数学模式中, 参数不含 CJK 字符, 可以在入口处直接关闭 XeTeXinterchartokenstate, 由数学模式的分组自动恢复。

```

7546 \__xeCJK_package_hook:nn { mTpro2 }
7547 {
7548 \cs_set_eq:NN \__xeCJK_mTpro_overbrace:w \overbrace
7549 \cs_gset_protected:Npn \overbrace #1
7550 {
7551 \makeXeCJKinactive
7552 \__xeCJK_mTpro_overbrace:w {#1}
7553 }
7554 \cs_set_eq:NN \__xeCJK_mTpro_underbrace:w \underbrace
7555 \cs_gset_protected:Npn \underbrace #1
7556 {

```

```

7557         \makexeCJKinactive
7558         \__xeCJK_mtpro_underbrace:w {#1}
7559     }
7560 }

```

__xeCJK_save_um_char: 兼容 unicode-math 和 CJKmath 选项, 避免将一些中西文混用的标点设置为 CJK 字体。

```

\__xeCJK_save_um_char:
7561 \__xeCJK_package_hook:nn { unicode-math }
7562 {
7563     \prop_const_from_keyval:Nn \c__xeCJK_um_ambiguous_char_prop
7564     {
7565         "00B7 = \cdotp ,
7566         "2025 = \enleadertwodots ,
7567         "2026 = \unicodeellipsis
7568     }
7569     \cs_new_protected:Npn \__xeCJK_save_um_char:
7570     {
7571         \cs_set_protected:Npe \__xeCJK_restore_um_char:
7572         {
7573             \prop_map_function:NN
7574             \c__xeCJK_um_ambiguous_char_prop
7575             \__xeCJK_restore_um_char_aux:nn
7576         }
7577     }
7578     \cs_new_eq:NN \__xeCJK_restore_um_char: \prg_do_nothing:
7579     \cs_new:Npn \__xeCJK_restore_um_char_aux:nn #1#2
7580     {
7581         \__xeCJK_gset_mathcodenum:nn
7582         { \int_value:w #1 }
7583         { \int_value:w \tex_Umathcodenum:D #1 }
7584     }
7585     \cs_new_protected:Npn \__xeCJK_gset_mathcodenum:nn #1#2
7586     {
7587         \int_compare:nNnF { \tex_Umathcodenum:D #1 } = {#2}
7588         { \tex_global:D \tex_Umathcodenum:D #1 = #2 ~ }
7589     }
7590 }

```

__xeCJK_patch_microtype_get_slot: 兼容 microtype。

```

7591 \cs_new_protected:Npn \__xeCJK_patch_microtype_get_slot:
7592 {
7593     \cs_new_eq:NN \xeCJK@original@get@slot \MT@get@slot@
7594     \cs_set_eq:NN \MT@get@slot@ \xeCJK@microtype@get@slot
7595     \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
7596 }
7597 \cs_new_protected_nopar:Npn \xeCJK@microtype@get@slot
7598 {
7599     \int_compare:nNnT \MT@char < \c_zero_int
7600     { \__xeCJK_get_ambiguous_slot: }
7601     \xeCJK@original@get@slot
7602 }
7603 \cs_new_protected:Npn \__xeCJK_get_ambiguous_slot:
7604 {
7605     \prop_get:NnNT \g__xeCJK_ambiguous_slot_prop
7606     { \MT@encoding - \tex_the:D \MT@toks } \l__xeCJK_tmp_t1
7607     { \cs_set_eq:NN \MT@char \l__xeCJK_tmp_t1 }
7608 }
7609 \cs_new_protected:Npn \xeCJK@microtype@restore@pickupfont
7610 { \__xeCJK_gadd_font_initial_hook:n { \MT@ltx@pickupfont } }
7611 \__xeCJK_package_hook:nn { microtype }
7612 {
7613     \cs_if_free:NF \MT@get@slot@
7614     { \__xeCJK_patch_microtype_get_slot: }
7615     \MT@addto@setup { \xeCJK@microtype@restore@pickupfont }
7616 }

```

简单处理与 `hyperref` 宏包的兼容问题。

`\Hy@BeginAnnot` 在 `annotation` 起始 `whatsit` 之前启动 `stream capture`; 链接文字中的 `Default/CJK` 转换随后直接记录实际首尾类别。顶层 `\Hy@endAnnot` 若看到数学节点, 而 `capture` 尚未由显式公式入口记录 `math`, 则把它作为 `URL` 等命令的可信 `Default` 输出记录下来; 显式 `\hyperref` 或 `\href` 参数以字面数学切换符开始或结束时, 由 `\hyper@link/\hyper@linkurl` 的三参数/两参数适配器分别报告首、末 `math` 类别。结束 `whatsit` 之后再结束 `capture`。这样普通西文、中文、混合链接文字、数学 `URL` 和可见行内公式都保留各自语义; 嵌套 `annotation` 仍只在最外层结束时收束, 不会让内部结束点提前重放状态。

```

7617 \cs_new_protected:Npn \__xeCJK_boundary_register_hyperref_annot:
7618 {
7619   \clist_map_inline:nn
7620   {
7621     Hy@BeginAnnot , Hy@endAnnot , hyper@link , hyper@linkurl ,
7622     href , hyperref
7623   }
7624   { \__xeCJK_boundary_reserve:n {##1} }
7625   \cs_gset_eq:NN \__xeCJK_orig_Hy_BeginAnnot:n \Hy@BeginAnnot
7626   \cs_gset_eq:NN \__xeCJK_orig_Hy_EndAnnot: \Hy@endAnnot
7627   \cs_gset_protected:Npn \Hy@BeginAnnot ##1
7628   {
7629     \__xeCJK_boundary_inline_stream_begin:n { auto }
7630     \__xeCJK_orig_Hy_BeginAnnot:n { ##1 }
7631   }
7632   \cs_gset_protected:Npn \Hy@endAnnot
7633   {
7634     \int_compare:nNt { \c@Hy@AnnotLevel } = { 1 }
7635     {
7636       \__xeCJK_if_last_math:T
7637       {
7638         \tl_if_eq:cnF
7639         {
7640           g__xeCJK_boundary_capture_
7641           \int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl
7642         }
7643         { math }
7644         { \__xeCJK_boundary_capture_class:n { default } }
7645       }
7646     }
7647     \__xeCJK_orig_Hy_EndAnnot:
7648     \__xeCJK_boundary_inline_stream_end:n { auto }
7649   }
7650   \cs_gset_eq:NN \__xeCJK_orig_hyper_link:nnn \hyper@link
7651   \cs_gset_protected:Npn \hyper@link ##1##2##3
7652   {
7653     \__xeCJK_orig_hyper_link:nnn {##1} {##2}
7654     {
7655       \__xeCJK_boundary_math_begin:n {##3}
7656       ##3
7657       \__xeCJK_boundary_math_end:n {##3}
7658     }
7659   }
7660   \cs_gset_eq:NN \__xeCJK_orig_hyper_linkurl:nn \hyper@linkurl
7661   \cs_gset_protected:Npn \hyper@linkurl ##1##2
7662   {
7663     \__xeCJK_orig_hyper_linkurl:nn
7664     {
7665       \__xeCJK_boundary_math_begin:n {##1}
7666       ##1
7667       \__xeCJK_boundary_math_end:n {##1}
7668     }
7669     {##2}
7670   }
7671 }
7672 \__xeCJK_package_hook:nn { hyperref }

```

```

7673 {
7674   \pdfstringdefDisableCommands
7675   {
7676     \__xeCJK_gobble_CJKfamily:
7677     \xeCJK_cs_clear:N \__xeCJK_inactive_group_begin:
7678     \xeCJK_cs_clear:N \__xeCJK_inactive_group_end:
7679     \xeCJK_cs_clear:N \makexeCJKinactive
7680     \xeCJK_cs_clear:N \xeCJK_text_composite_patch:
7681   }
7682   \__xeCJK_boundary_register_hyperref_annot:
7683 }

```

color/xcolor 的颜色切换通过 \set@color 插入颜色 push whatsit, 再由 aftergroup 中的 \reset@color 插入 pop whatsit。两者都没有可见输出, 因此分别注册为 transparent: 入口保存并移除 xeCJK marker 与源码空格, 颜色 whatsit 排出后原样恢复。 \color@b@x 会直接写出多个节点, 继续使用 wrapped-box 收集整个盒子命令的输出。 \@textcolor 的当前三参数内部签名由一层 stream 适配器保留; 第三个参数以字面数学切换符开始或结束时, 适配器分别在 math-on 前报告首类别、在正文参数实际排完时确认末类别。

```

7684 % \tn{colorbox} 和 \tn{fcolorbox} 的扫描器都会把可见正文留作
7685 % \tn{color@b@x} 的第三个参数。先包装这个共同入口, 再安装 wrapped-box
7686 % hook; 调用时便会先启动 capture, 再由适配器报告正文两端的公式类别。
7687 % \pkg{xcolor} 会重定义这个私有函数, 因此它的 package hook 需要按
7688 % 当前定义重新保存原函数。
7689 \cs_new_protected:Npn \__xeCJK_boundary_color_box:nnn #1#2#3
7690 {
7691   \__xeCJK_boundary_math_begin:n {#3}
7692   \__xeCJK_orig_color_box:nnn {#1} {#2}
7693   { #3 \__xeCJK_boundary_math_end:n {#3} }
7694 }
7695 \cs_new_protected:Npn \__xeCJK_boundary_prepare_color_box:
7696 {
7697   \__xeCJK_boundary_reserve:n { color@b@x }
7698   \cs_if_eq:NNF \color@b@x \__xeCJK_boundary_color_box:nnn
7699   {
7700     \cs_gset_eq:NN \__xeCJK_orig_color_box:nnn \color@b@x
7701     \cs_gset_eq:NN \color@b@x \__xeCJK_boundary_color_box:nnn
7702   }
7703 }
7704 \cs_new_protected:Npn \__xeCJK_boundary_textcolor:nnn #1#2#3
7705 {
7706   \__xeCJK_boundary_if_math_edge:nTF {#3}
7707   {
7708     \__xeCJK_boundary_inline_stream_begin:n { auto }
7709     \__xeCJK_boundary_math_begin:n {#3}
7710     \__xeCJK_orig_textcolor:nnn {#1} {#2}
7711     { #3 \__xeCJK_boundary_math_end:n {#3} }
7712     \__xeCJK_boundary_inline_stream_end:n { auto }
7713   }
7714   { \__xeCJK_orig_textcolor:nnn {#1} {#2} {#3} }
7715 }
7716 \cs_new_protected:Npn \__xeCJK_boundary_register_textcolor:
7717 {
7718   \__xeCJK_boundary_reserve:n { @textcolor }
7719   \__xeCJK_boundary_reserve:n { textcolor }
7720   \cs_if_eq:NNF \@textcolor \__xeCJK_boundary_textcolor:nnn
7721   {
7722     \cs_gset_eq:NN \__xeCJK_orig_textcolor:nnn \@textcolor
7723     \cs_gset_eq:NN \@textcolor \__xeCJK_boundary_textcolor:nnn
7724   }
7725 }
7726 \cs_new_protected:Npn \__xeCJK_boundary_register_color:
7727 {
7728   \__xeCJK_boundary_reserve:n { colorbox }
7729   \__xeCJK_boundary_reserve:n { fcolorbox }
7730   \__xeCJK_boundary_register_transparent:n { set@color }

```

```

7731 \__xeCJK_boundary_register_transparent:n { reset@color }
7732 \__xeCJK_boundary_prepare_color_box:
7733 \__xeCJK_boundary_register_wrapped_box:nn { color@b@x } { auto }
7734 \__xeCJK_boundary_register_textcolor:
7735 }
7736 \__xeCJK_package_hook:nn { color }
7737 { \__xeCJK_boundary_register_color: }
7738 \__xeCJK_package_hook:nn { xcolor }
7739 { \__xeCJK_boundary_register_color: }

```

`\l3color`(`expl3` 内置)的颜色机制使用独立的后端代码路径,不经过`\set@color/\reset@color`。`\__color_select:N`负责颜色推入(调用后端 `select` 并注册 `aftergroup reset`)，`\__color_backend_reset:`负责颜色弹出。两者都没有可见输出,使用与 `LaTeX2e color push/pop` 相同的 `transparent capture`。`expl3` 内部函数不接受通用 `cmd hook`, 因此只用保持原参数签名的薄适配器包围 `framework begin/end`, 不再复制 `marker` 检测与重放算法。

```

7740 \cs_new_protected:Npn \__xeCJK_boundary_register_l3color:
7741 {
7742   \__xeCJK_boundary_wrap_transparent_onearg:NN
7743   \__xeCJK_orig_color_select:N \__color_select:N
7744   \__xeCJK_boundary_wrap_transparent_noarg:NN
7745   \__xeCJK_orig_color_backend_reset: \__color_backend_reset:
7746 }
7747 \__xeCJK_boundary_register_l3color:

```

`hypdoc` 的 `\HD@target` 通过 `\raisebox` 产生 `hbox(0+0)x0` 用于放置 `\special{pdf:dest...}`。该 `hbox` 隔在 `xeCJK` 标记 `kern` 对和后续字符之间,使`\tex_lastkern:D`探测失败,导致 `Boundary→{Default,CJK}` 路径中`\xeCJK_if_last_node:n / \@@_if_last_kern:`检查返回 `false`, 间距因此丢失。该命令没有可见输出,因此注册为 `transparent`: 入口保存并移除 `marker` 与可选源码空格,原命令插入锚点 `hbox` 后再原样恢复;后续边界不再受这个 `0×0 hbox` 遮蔽。

```

7748 \__xeCJK_package_hook:nn { hypdoc }
7749 { \__xeCJK_boundary_register_transparent:n { HD@target } }

```

`siunitx` 用 `math` 模式排版数字与单位。`math` 不触发 `interchar` 转换,入口的 `\mathon` 又盖住左侧 `marker`, 情况与修复前的 `\eqref` 相同;数字与单位输出必然以西文或数字开始、结束,因此也注册为固定 `Default` 首尾的 `stream capture`。`v2` 旧名 `\si`、`\SI` 是独立顶层命令,不经由 `\unit`、`\qty` 的 `cmd hook`, 需单独注册。注册前先检查命令是否存在,以兼容未来移除旧名的 `siunitx` 版本。若其他宏包恰好定义了同名 `\si`、`\SI`, 这项检查也会把它们一并注册——误注册只是多包一层 `Default stream`, 不改变无 `CJK` 邻接时的排版,记录为已知限制。数学模式内使用(如 `$x=\unit{kg}$`)时, `capture` 只压入 `inactive` 标记,不改变排版。`\ang` 输出角度符号,目前还没有确定应与哪种直接输入比较,因此暂不注册。

```

7750 \cs_new_protected:Npn \__xeCJK_boundary_register_siunitx:
7751 {
7752   \__xeCJK_boundary_register_stream:nn { unit } { default }
7753   \__xeCJK_boundary_register_stream:nn { qty } { default }
7754   \__xeCJK_boundary_register_stream:nn { num } { default }
7755   \cs_if_exist:NT \si
7756   { \__xeCJK_boundary_register_stream:nn { si } { default } }
7757   \cs_if_exist:NT \SI
7758   { \__xeCJK_boundary_register_stream:nn { SI } { default } }
7759 }
7760 \__xeCJK_package_hook:nn { siunitx }
7761 { \__xeCJK_boundary_register_siunitx: }

```

`url` 的 `\url` 命令通过 `\Url@FormatString` 进入数学模式排版 `URL` 字符串。数学模式 `node` 隔在 `xeCJK` 标记 `kern` 对和数学公式之间, 又因 `X3TεX` 在数学模式中不触发类别转换 (`interchar class`), 标记 `kern` 对永远等不到匹配的 `Boundary→{CJK,Default}` 触发, 缓存的 `CJKecglue` 因此在 `CJK` 与 `\url` 之间丢失。`\url` 同时支持花括号和任意分隔符扫描, 不能用普通 `cmd/url/before / cmd/url/after hook` 包围; `after hook` 会进入分隔符参数本身。`\Url@z`

已经完成参数扫描,并覆盖 `\Url@HyperHook`、数学格式化和最后的分组关闭,因此在这里启动和结束固定 Default 首尾的 stream capture。数学模式不会触发 interchar 转换,但 default 模式会显式报告两端类别;URL 左右边界由统一框架重建,不再需要专用的 marker 移除与补胶逻辑。

```

7762 \cs_new_protected:Npn \__xeCJK_boundary_register_url:
7763 {
7764   \clist_map_inline:nn { url , path , nolinkurl , Url@z }
7765     { \__xeCJK_boundary_reserve:n {##1} }
7766   \cs_if_exist:NF \__xeCJK_orig_Url_z:n
7767     {
7768       \cs_set_eq:NN \__xeCJK_orig_Url_z:n \Url@z
7769       \cs_gset_protected:Npn \Url@z ##1
7770         {
7771           \__xeCJK_boundary_inline_stream_begin:n { default }
7772           \__xeCJK_orig_Url_z:n {##1}
7773           \__xeCJK_boundary_inline_stream_end:n { default }
7774         }
7775     }
7776 }
7777 \__xeCJK_package_hook:nn { url } { \__xeCJK_boundary_register_url: }

```

$\text{\LaTeX} 2_{\epsilon}$ 的 `\verb` 命令展开后会执行 `\leavevmode`, 其中 `\null` 等价于 `\hbox{}`, 会产生一个 `\hbox(0+0)x0` 节点。这个 0×0 的 `\hbox` 紧跟在 `xeCJK` 的标记 `kern` 对之后, 会切断后续 `Boundary` $\rightarrow$ `{Default, CJK}` 类别转换中 `\tex_lastkern:D` 的探测路径, 导致 `CJK` 文字与 `\verb` 之间应有的 `CJKecglue` 丢失。

原 #910 补丁在入口直接补 `\CJKecglue`, 隐含假设 `verbatim` 内容必为西文, 因而不能处理中文输出或右边界。现在在原始 `\verb` 执行前启动 auto stream capture: 入口先保存外侧 marker、源码空格和 glue 值; 分隔符扫描及排版仍完全交给内核。`\verb@egroup` 关闭组并刷新 language whatsit 后结束 capture, 按真正观察到的首尾 Default/CJK 类别同时重建左右边界。数学模式由 capture 入口自动跳过; `shortverb` 的短记号与 `\verb*` 共用同一入口和组结束点, 因此继续自动覆盖。

补丁仍在 `\__xeCJK_after_preamble:n` 安装, 以避免其他宏包在导言区重定义 `\verb` 的时序。

`\verb` 右侧还有一处独立时序(#919): `\verb` 组内把 `\language` 切到 `\l@nohyphenation`, 组关闭后的 language whatsit 会延迟到下一字符进入列表时才排出。若让它留到 capture 结束之后, 它会遮住 framework 刚刚重放的末尾 marker。这里仍在 `\verb@egroup` 关组后立即执行 `\setlanguage\language`, 把 language whatsit 收进 stream capture; 随后 framework 在它之后重放实际末类别, 右侧有、无源码空格都交给普通恢复链处理。旧的通用 whatsit 类别猜测已经删除, 不再参与这条路径。

`\__xeCJK_flush_language_whatsit:` 按当前 `\language` 值主动排出一个 language whatsit。`\setlanguage\language` 只排出 whatsit 记录当前值, 不改变 hyphenation 状态, 对断行无副作用。仅在外层水平模式下执行: 数学模式或行首场景没有上述时序问题, restricted horizontal mode (`\hbox` 内) 则根本不产生延迟 whatsit。目前唯一调用点是 `\verb@egroup` (见下方 `\__xeCJK_boundary_register_verb:`); 未来若有其他切换 `\language` 的分组宏需要同样的时序修正, 可直接复用本函数, 并在此注释中登记新的调用点。

```

7778 \cs_new_protected:Npn \__xeCJK_flush_language_whatsit:
7779 { \mode_if_horizontal:T { \tex_setlanguage:D \tex_language:D } }

```

`shortverb` 短记号与 `\verb*` 都经由 `\verb` 入口且组关闭走同一个 `\verb@egroup`, 因此 capture 与 flush 两个补丁都自动覆盖。

```

\__xeCJK_boundary_register_verb: 7780 \cs_new_protected:Npn \__xeCJK_boundary_register_verb:
7781 {
7782   \cs_if_exist:NF \__xeCJK_orig_verb:
7783   {

```

```

7784 \cs_set_eq:NN \__xeCJK_orig_verb: \verb
7785 \cs_gset_protected:Npn \verb
7786 {
7787   \__xeCJK_boundary_inline_stream_begin:n { auto }
7788   \__xeCJK_orig_verb:
7789 }
7790 \tl_gput_right:Nn \verb@egroup
7791 {
7792   \__xeCJK_flush_language_whatsit:
7793   \__xeCJK_boundary_inline_stream_end:n { auto }
7794 }
7795 }
7796 }
7797 \__xeCJK_boundary_reserve:n { verb }
7798 \__xeCJK_boundary_reserve:n { verb@egroup }
7799 \__xeCJK_after_preamble:n { \__xeCJK_boundary_register_verb: }

```

`\l3doc` 类用 `\__codeloc_meta:n` 排版 `\meta` 命令的参数：内部由 `\__codeloc_meta-original:n` 依次输出 `\ensuremath{\langle \rangle}` 数学节点、参数内容、以及对称的 `\ensuremath{\langle \rangle}`。当参数内容的首字符是 CJK 时，`Default→CJK` 类别转换中的 `\xeCJK_pre_inter_class_toks:nnn` 会自动加上 `\CJKe glue`，这会在 `<` 与首个 CJK 字符之间产生多余的间距。

这里采用 myhsia 在 `hyperref / hypdoc` 兼容议题中提出的 `hbox` 包装方案：把 `\__codeloc_meta:n` 的参数装进 `\hbox:n`。 `hbox` 隔离了内部的 $\TeX$ 类别转换，`Default→CJK` 转换不会跨越 `hbox` 边界触发，从而避免插入多余的 `\CJKe glue`。

仅用 `hbox` 包装会让入口的 `math` 节点遮住左侧 marker。当前适配器因此同时用固定 `Default` 首尾的 `stream capture` 包围完整 `\__codeloc_meta:n: framework` 负责外侧两端，`hbox` 只负责尖括号内部的输出规范化。这样 `\meta`、`\Arg`、`\oarg`、`\parg` 等所有共用内部实现的入口都走同一恢复算法，不再保留专用 `drain`。

`\__codeloc_meta:n` 同时被 `\Arg`、`\oarg`、`\parg` 等命令共用，此适配器也覆盖后者的相同问题。

这里选择 `\__xeCJK_at_end_preamble:n` 而非 `\__xeCJK_package_hook:nn`，原因在于 `\l3doc` 是文档类，文档类加载早于 `xeCJK`，需要等到 `preamble` 结束时才能判定 `\__codeloc_meta:n` 是否定义。同时亦有开发者可能会在 `xeCJK` 前加载 `doc` 宏包。

```

\__xeCJK_boundary_register_codedoc_meta: 7800 \cs_new_protected:Npn \__xeCJK_boundary_register_codedoc_meta:
7801 {
7802   \cs_if_exist:NTF \__codeloc_meta:n
7803   {
7804     \clist_map_inline:nn
7805     { \__codeloc_meta:n , meta , Arg , marg , oarg , par }
7806     { \__xeCJK_boundary_reserve:n {##1} }
7807     \cs_if_exist:NF \__xeCJK_orig_codedoc_meta:n
7808     {
7809       \cs_set_eq:NN \__xeCJK_orig_codedoc_meta:n \__codeloc_meta:n
7810       \cs_gset_protected:Npn \__codeloc_meta:n ##1
7811       {
7812         \__xeCJK_boundary_inline_stream_begin:n { default }
7813         \__xeCJK_orig_codedoc_meta:n { \hbox:n {##1} }
7814         \__xeCJK_boundary_inline_stream_end:n { default }
7815       }
7816     }
7817   }
7818   {
7819     \bool_lazy_and:nnT
7820     { \cs_if_exist_p:c { ver@doc.sty } }
7821     { \cs_if_free_p:N \__xeCJK_orig_codedoc_meta:n }
7822     {
7823       \__xeCJK_boundary_reserve:n { meta }
7824       \cs_set_eq:NN \__xeCJK_orig_codedoc_meta:n \meta
7825       \DeclareDocumentCommand \meta { m }
7826       {
7827         \__xeCJK_boundary_inline_stream_begin:n { default }

```

```

7828         \__xeCJK_orig_codedoc_meta:n { \hbox:n {##1} }
7829         \__xeCJK_boundary_inline_stream_end:n { default }
7830     }
7831 }
7832 }
7833 }
7834 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_register_codedoc_meta: }
7835 % \changes{v3.10.4}{2026/07/20}{将 \tn{eqref} 注册为固定 Default
7836 % 首尾、将 \tn{cs} 注册为仅固定 Default 首端的 stream capture;
7837 % codedoc/doc 的 meta 内部适配器直接包围固定 Default stream, 删除专用
7838 % drain 并覆盖其所有公共调用方 (\#992)。}
7839 % \changes{v3.10.4}{2026/07/20}{将内核 \tn{@setref} (或 \pkg{hyperref}
7840 % 保存的 \tn{real@setref}) 注册为 auto stream capture, 以统一框架取代
7841 % \#991 的专用 saved-node/replay 补丁。}
7842 % \tn{eqref} 的圆括号和 \tn{meta} 的数学尖括号决定了命令两端
7843 % 始终是 Default; 参数内部即使包含中文, 也不能把外侧 oracle
7844 % 误判为裸中文。 \tn{cs} 只有开头的反斜线固定为 Default, 末尾仍由
7845 % 参数的实际最后字符决定, 因此使用 first-default stream 固定首类别并记录
7846 % 真实末类别。
7847 \__xeCJK_at_end_preamble:n
7848 {
7849     \cs_if_exist:NTF \real@setref
7850     { \__xeCJK_boundary_register_stream:nn { real@setref } { auto } }
7851     { \__xeCJK_boundary_register_stream:nn { @setref } { auto } }
7852     \cs_if_exist:NTF \eqref
7853     { \__xeCJK_boundary_register_stream:nn { eqref } { default } }
7854     \cs_if_exist:NTF \cs
7855     { \__xeCJK_boundary_register_stream:nn { cs } { first-default } }
7856 }

```

`biblatex` 在开启 `pagetracker` 选项 (`authoryear` 等 `bbx` 样式默认开启) 时, 每次排版 `\bibitem` 会在 `\blx@pagetracker@context` 内调用 `\protected@write` 往 `aux` 写入 `\abx@aux@page` 记录。该 `\write` 产生的 `whatsit` 节点会遮蔽前一条目结尾的 `marker`。历史通用 `whatsit` 回退曾把可能陈旧的 `\g__xeCJK_last_node_tl` 当作列表证据, 因而在下一条目首字符前误加 `CJKecglue`; 该通用猜测现已删除。

#931 的旧补丁只清空 `\g__xeCJK_last_node_tl`, 能阻止虚假 `ecglue`, 却也丢掉了 `whatsit` 两侧本来应有的 `CJKecglue/CJKglue`。 `pagetracker` 本身没有可见输出, 因此现在保存入口 `marker`、源码空格和 `glue` 状态, 执行 `write` 后原样重放; 已知 `whatsit` 不再遮蔽真实边界, 也不会把陈旧状态带入下一个无关段落。

注册点选 `\blx@pagetracker` 而非 `\blx@pagetracker@context`: `biblatex` 在 `bbx style` 加载时通过 `\ExecuteBibliographyOptions{pagetracker}` 展开 `\blx@opt@pagetracker@page`, 其内部 `\let \blx@pagetracker←\blx@pagetracker@context`——注意是 `**\let` 值传递 `**` 而非 `**` 符号引用 `**`。若补丁挂在 `\blx@pagetracker@context`, 只能改到源函数; 但 `\blx@bibitem` 实际调用的是 `\blx@pagetracker` 的 `\let` 拷贝, 早在 `bbx` 加载时就已定型, 之后再改源函数已经无效。故补丁点选 `\blx@pagetracker` 本身, 并延迟到 `\__xeCJK_at_end_preamble:n` 执行——此时 `bbx` 已完全加载, `\blx@pagetracker` 已 `\let` 到最终目标 (可能是 `\blx@pagetracker@context`、`\blx@pagetracker@spread` 或 `\relax`)。

```

\__xeCJK_boundary_register_biblatex_pagetracker: 7857 \cs_new_protected:Npn \__xeCJK_boundary_register_biblatex_pagetracker:
7858 {
7859     \cs_if_exist:NTF \blx@pagetracker
7860     { \__xeCJK_boundary_register_transparent:n { blx@pagetracker } }
7861 }
7862 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_register_biblatex_pagetracker: }
7863 % 用户声明在全部内建 preamble 末注册之后应用; 普通命令 hook 仍由 LaTeX
7864 % 延迟到 \tn{AtBeginDocument} 代码执行完再安装。随后检查目标命令是否存在,
7865 % 避免不存在的控制序列被 LaTeX 静默跳过。
7866 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_user_register_apply: }
7867 \__xeCJK_after_end_preamble:n { \__xeCJK_boundary_user_register_check: }

```

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

```
7868 \__xeCJK_package_hook:nn { cprotect }
7869 {
7870   \cs_if_free:NF \icprotect
7871   { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
7872 }
```

在 `listings` 宏包后自动载入 `xeCJK-listings`。

```
7873 \__xeCJK_package_hook:nn { listings }
7874 { \RequirePackage { xeCJK-listings } }
```

由于 `xeCJK` 假装 `CJK` 已经被引入了,这会可能导致旧版本的 `everyysel` 包判断错误。需要在它们判断之前取消定义。

```
7875 \__xeCJK_package_hook:nn { everyysel }
7876 {
7877   \cs_if_exist:NF \@EverySelectfont@Legacy
7878   { \cs_undefine:c { ver@CJK . \c__xeCJK_package_ext_tl } }
7879 }
```

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

```
7880 \ctex_at_begin_package:nn { CJKnumb }
7881 {
7882   \tl_new:N \l__xeCJK_CJK_version_tl
7883   \tl_set_eq:Nc \l__xeCJK_CJK_version_tl { ver@CJK . \c__xeCJK_package_ext_tl }
7884   \tl_set:cn { ver@CJK . \c__xeCJK_package_ext_tl } { 9999/99/99 }
7885   \cs_new_protected:Npn \CJKaddEncHook #1#2
7886   {
7887     \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
7888     {
7889       \group_begin:
7890       \cs_set_eq:NN \Unicode \xeCJK_unicode_char:nn
7891       \cs_set_eq:NN \def \xdef
7892       #2
7893       \group_end:
7894       \str_gset:Nn \CJK@tenthousand { ~~~~4e07 }
7895       \str_gset:Nn \CJK@hundredmillion { ~~~~4ebf }
7896       \tl_if_exist:NF \CJK@UnicodeEnc
7897       { \tl_const:Nn \CJK@UnicodeEnc { UTF8 } }
7898       \cs_if_exist:NF \Unicode
7899       { \cs_new_eq:NN \Unicode \xeCJK_unicode_char:nn }
7900     }
7901   }
7902   \cs_new:Npn \xeCJK_unicode_char:nn #1#2
7903   { \tex_Uchar:D \tex_numexpr:D (#1) * 256 + (#2) \scan_stop: }
7904 }
7905 \ctex_at_end_package:nn { CJKnumb }
7906 { \tl_set_eq:cn { ver@CJK . \c__xeCJK_package_ext_tl } \l__xeCJK_CJK_version_tl }
```

最后引入本地配置文件。

```
7907 \bool_if:NT \g__xeCJK_config_bool
7908 {
7909   \ExplSyntaxOff
7910   \file_input:n { \g__xeCJK_config_name_tl .cfg }
7911   \ExplSyntaxOn
7912 }
7913 </package>
```

5.19 xeCJKfntef

```

7914 <fntef>

7915 \PassOptionsToPackage { normalem } { ulem }
7916 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
7917 \ProcessOptions \scan_stop:

7918 \RequirePackage { xeCJK }
7919 \RequirePackage { ulem }

7920 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

\xeCJK_hook_for_ulem: 7921 \cs_new_protected:Npn \xeCJK_hook_for_ulem:
7922 {
7923   \xeCJK_ulem_detect_node:
7924   \l__xeCJK_ulem_text_format_tl
7925   \bool_if:NF \l__xeCJK_ulem_hook_used_bool
7926   {
7927     \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
7928     \__xeCJK_ulem_hook:
7929   }
7930   \xeCJK_ulem_begin_node:
7931 }
7932 \cs_new_protected:Npn \__xeCJK_ulem_hook:
7933 {
7934   \__xeCJK_ulem_initial:
7935   \bool_if:NT \l__xeCJK_ulem_subtract_bool
7936   {
7937     \xeCJK_swap_cs:NN \UL@leaders \xeCJK_ulem_leaders:
7938     \cs_set_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:
7939     \cs_set_eq:NN \xeCJK_ulem_right_skip: \__xeCJK_ulem_right_skip:
7940   }
7941   \bool_if:NT \l__xeCJK_ulem_hidden_bool
7942   { \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_hidden_box: }
7943   \bool_if:NTF \l__xeCJK_ulem_skip_bool
7944   {
7945     \cs_set_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
7946     \cs_set_eq:NN \__xeCJK_ulem_hskip_aux:n \xeCJK_ulem_hskip:n
7947   }
7948   {
7949     \xeCJK_swap_cs:NN \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
7950     \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
7951     \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
7952   }
7953   \xeCJK_glue_to_skip:nN
7954   {
7955     \cs_set_eq:NN \ \tex_space:D
7956     \cs_set_eq:NN \penalty \tex_penalty:D
7957     \cs_set_eq:NN \hskip \skip_horizontal:N
7958     \CJKglue
7959   } \l__xeCJK_ccglue_skip
7960   \xeCJK_glue_to_skip:nN
7961   {
7962     \cs_set_eq:NN \ \tex_space:D
7963     \cs_set_eq:NN \penalty \tex_penalty:D
7964     \cs_set_eq:NN \hskip \skip_horizontal:N
7965     \CJKecglue
7966   } \l__xeCJK_ecglue_skip
7967   \xeCJK_glue_to_skip:nN { \xeCJK_space_glue: } \l__xeCJK_space_skip
7968   \cs_set_protected:Npn \CJKglue
7969   { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
7970   \cs_set_protected:Npn \CJKecglue
7971   { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
7972   \cs_set_protected:Npn \xeCJK_space_glue:
7973   { \__xeCJK_ulem_glue:n \l__xeCJK_space_skip }
7974   \cs_set_eq:NN \xeCJK_punct_node:N \use_none:n
7975   \cs_set_eq:NN \xeCJK_if_last_punct:TF \use_ii:nn
7976   \keys_set:nn { xeCJK / options }

```

```

7977     { CheckFullRight = false , xCJKecglue = false }
7978   }
7979   \skip_new:N \l__xeCJK_space_skip
7980   \bool_new:N \l__xeCJK_ulem_hook_used_bool

```

\UL@word 修改 \UL@word, 目的是取得分组中的 \UL@leadtype, 以便加入 \xeCJK_ulem_right_skip:。

```

\__xeCJK_ulem_word:nw
7981 \clist_map_inline:nn
7982 {
7983   UL@word , ULon , UL@on , UL@onin ,
7984   uuline , uuline , uwave , sout , xout , dashuline , dotuline ,
7985   xeCJKfntefon , CJKunderline , CJKunderwave , CJKunderdblline ,
7986   CJKsout , CJKxout , CJKunderanyline , CJKunderanysymbol , CJKunderdot
7987 }
7988 { \__xeCJK_boundary_reserve:n {#1} }
7989 \cs_new_protected:Npn \xeCJK_ulem_word:nw #1 ~
7990 {
7991   \__xeCJK_ulem_start:w #1 ~
7992   \exp_after:wN \if_meaning:w \exp_after:wN \UL@end #1
7993   \exp_after:wN \__xeCJK_ulem_end:
7994   \else:
7995     \exp_after:wN \__xeCJK_ulem_loop:nw
7996   \fi:
7997 }
7998 \cs_new_protected:Npn \__xeCJK_ulem_end:
7999 {
8000   \c_group_end_token
8001   \c_group_end_token
8002   \tex_unskip:D \tex_unskip:D \tex_unskip:D
8003   \xeCJK_ulem_right_skip:
8004   \xeCJK_ulem_group_end:
8005   \xeCJK_ulem_right_node:
8006   \__xeCJK_boundary_inline_stream_end:n { auto }
8007   \bool_set_false:N \l__xeCJK_ulem_stream_started_bool
8008   \int_set:Nn \tex_spacefactor:D { \UL@spfactor }
8009 }
8010 \cs_new_protected:Npn \__xeCJK_ulem_loop:nw
8011 {
8012   \reverse_if:N \if_mode_math:
8013   \reverse_if:N \if_dim:w \tex_lastskip:D = \c_zero_dim
8014   \skip_gset_eq:NN \UL@skip \tex_lastskip:D
8015   \tex_unskip:D
8016   \UL@stop \UL@leaders
8017   \fi:
8018   \fi:
8019   \xeCJK_ulem_word:nw \prg_do_nothing:
8020 }
8021 \cs_new_protected:Npn \__xeCJK_ulem_start:w
8022 { \exp_after:wN \UL@start }
8023 \cs_set_eq:NN \UL@word \xeCJK_ulem_word:nw

```

\xeCJK_ulem_left: 在下划线开始之前探测之前的 node, 以便随后插入 \CJKglue 或 \CJKecglue。

```

\__xeCJK_ulem_detect_node:
8024 \cs_new_protected:Npn \xeCJK_ulem_left:
8025 {
8026   \xeCJK_ulem_left_node:
8027   \xeCJK_make_group_tag:
8028 }
8029 \cs_new_eq:NN \xeCJK_ulem_left_node: \prg_do_nothing:
8030 \cs_new_protected:Npn \xeCJK_ulem_detect_node:
8031 {
8032   \scan_stop:
8033   \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
8034   {
8035     \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
8036     \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
8037     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8038   }

```

```

8039 {
8040   \dim_set_eq:NN \l__xeCJK_tmp_dim \tex_lastkern:D
8041   \tex_unkern:D
8042   \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_tmp_dim }
8043   {
8044     \tex_unkern:D
8045     \cs_set_protected:Npe \xeCJK_ulem_left_node:
8046     {
8047       \tex_kern:D - \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
8048       \tex_kern:D \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
8049     }
8050     \cs_set_protected:Npn \xeCJK_ulem_begin_node:
8051     { { \xeCJK_make_node:n { ulem-begin } } }
8052     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
8053   }
8054   {
8055     \tex_kern:D \l__xeCJK_tmp_dim
8056     \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
8057     \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
8058     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8059   }
8060 }
8061 }
8062 \xeCJK_declare_node:n { ulem-begin }
8063 \cs_new_eq:NN \xeCJK_ulem_begin_node: \prg_do_nothing:

```

`\__xeCJK_ulem_hskip_first:n` 如果第一次调用的 `\CJKglue` 或 `\CJKecglue` 由下划线中的第一个文字和之前的内容产生, 就
`\xeCJK_ulem_hskip:n` 不用画下划线。

```

8064 \cs_new_protected:Npn \__xeCJK_ulem_hskip_first:n #1
8065 {
8066   \xeCJK_if_last_node:nTF { ulem-begin }
8067   {
8068     \xeCJK_remove_node:
8069     \skip_horizontal:n {#1}
8070   }
8071   { \xeCJK_ulem_hskip:n {#1} }
8072   \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8073 }
8074 \cs_new_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
8075 \cs_new_protected:Npn \xeCJK_ulem_hskip:n #1
8076 { { \skip_set:Nn \UL@skip {#1} \UL@leaders } }

```

`\xeCJK_ulem_right:` 从 ulem 内部列表取出末尾 marker, 并在外层列表原样放回; stream capture 结束时会把这个
`\xeCJK_ulem_right_node:` 可信 marker 解释为实际末类别。

```

8077 \cs_new_protected:Npn \xeCJK_ulem_right:
8078 {
8079   \scan_stop:
8080   \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
8081   { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
8082   {
8083     \dim_compare:nNnTF \tex_lastkern:D = { 3sp }
8084     { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
8085     {
8086       \exp_args:NNo \tex_unkern:D
8087       \__xeCJK_ulem_right_aux:n { \dim_use:N \tex_lastkern:D }
8088     }
8089   }
8090 }
8091 \cs_new_protected:Npn \__xeCJK_ulem_right_aux:n #1
8092 {
8093   \dim_compare:nNnTF \tex_lastkern:D = { - #1 }
8094   {
8095     \tex_unkern:D
8096     \cs_gset_protected:Npn \xeCJK_ulem_right_node:
8097     {

```

```

8098         \tex_kern:D - #1 \exp_stop_f:
8099         \tex_kern:D #1 \exp_stop_f:
8100     }
8101     \tl_gset:Ne \UL@spfactor { \int_use:N \tex_spacefactor:D }
8102 }
8103 {
8104     \tex_kern:D #1 \exp_stop_f:
8105     \xeCJK_cs_gclear:N \xeCJK_ulem_right_node:
8106 }
8107 }
8108 \cs_new_eq:NN \xeCJK_ulem_right_node: \prg_do_nothing:

```

\xeCJK_ulem_var_leaders: 第一次画下划线时, 不需要向左平移 \UL@pixel, 让左侧有间距。

```

8109 \cs_new_protected:Npn \xeCJK_ulem_leaders:
8110 { \__xeCJK_ulem_var_leaders: }
8111 \cs_new_protected:Npn \xeCJK_ulem_var_leaders:
8112 {
8113     \scan_stop:
8114     \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8115     {
8116         \UL@leadtype \skip_horizontal:n { \UL@skip + \UL@pixel }
8117         \skip_horizontal:n { - \UL@pixel }
8118         \cs_gset_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_leaders:
8119     }
8120 }
8121 \cs_new_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:

```

\xeCJK_ulem_right_skip: 在下划线完全画好之后, 我们检测最后的情况。用 \unskip 去掉最后一个下划线, 再重新画一个减少 \UL@pixel 的。

```

8122 \cs_new_eq:NN \xeCJK_ulem_right_skip: \prg_do_nothing:
8123 \cs_new_protected:Npn \__xeCJK_ulem_right_skip:
8124 {
8125     \int_case:nn { \tex_lastnodetype:D }
8126     {
8127         { \c__xeCJK_hlist_node } { \__xeCJK_ulem_right_skip_hbox: }
8128         { \c__xeCJK_glue_node } { \__xeCJK_ulem_right_skip_glue: }
8129         { \c__xeCJK_penalty_node } { \__xeCJK_ulem_right_skip_penalty: }
8130     }
8131 }
8132 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_hbox:
8133 {
8134     \box_set_to_last:N \l__xeCJK_tmp_box
8135     \__xeCJK_if_last_kern:TF
8136     { \__xeCJK_ulem_right_skip_kern: }
8137     { \__xeCJK_ulem_right_skip_glue: }
8138     \box_use_drop:N \l__xeCJK_tmp_box
8139 }
8140 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_kern:
8141 {
8142     \dim_set:Nn \l__xeCJK_tmp_dim { - \box_wd:N \l__xeCJK_tmp_box }
8143     \dim_compare:nNt \tex_lastkern:D = \l__xeCJK_tmp_dim
8144     {
8145         \tex_unkern:D
8146         \__xeCJK_ulem_right_skip_glue:
8147         \tex_kern:D \l__xeCJK_tmp_dim
8148     }
8149 }
8150 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_glue:
8151 {
8152     \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8153     {
8154         \tex_unskip:D
8155         \skip_set:Nn \l__xeCJK_tmp_skip { \tex_lastskip:D - \UL@pixel }
8156         \tex_unskip:D
8157         \UL@leadtype \skip_horizontal:N \l__xeCJK_tmp_skip
8158     }

```

```

8159 }
8160 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_penalty:
8161 {
8162   \int_set_eq:NN \l__xeCJK_tmp_int \tex_lastpenalty:D
8163   \tex_unpenalty:D
8164   \__xeCJK_if_last_hlist:T
8165   { \__xeCJK_ulem_right_skip_hbox: }
8166   \tex_penalty:D \l__xeCJK_tmp_int
8167 }

```

__xeCJK_ulem_hidden_box: 只画线, 不输出盒子。

```

8168 \cs_new_protected:Npn \__xeCJK_ulem_hidden_box:
8169 {
8170   \tl_if_empty:NF \UL@start
8171   {
8172     \box_set_ht:Nn \l__xeCJK_hidden_box { \box_ht:N \UL@box }
8173     \box_set_dp:Nn \l__xeCJK_hidden_box { \box_dp:N \UL@box }
8174     \box_use:N \l__xeCJK_hidden_box
8175     \xeCJK_no_break:
8176     \xeCJK_ulem_hskip:n { \box_wd:N \UL@box }
8177     \box_use:N \l__xeCJK_hidden_box
8178   }
8179 }
8180 \box_new:N \l__xeCJK_hidden_box
8181 \hbox_set:Nn \l__xeCJK_hidden_box { }

```

__xeCJK_ulem_skip_punct_begin: 让下划线跳过标点符号的设置。

__xeCJK_ulem_skip_punct_end:

```

8182 \cs_new_protected:Npn \__xeCJK_ulem_skip_punct_begin:
8183 {
8184   \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_skip_putbox:
8185   \cs_set_eq:NN \xeCJK_ulem_hskip:n \skip_horizontal:n
8186 }
8187 \cs_new_protected:Npn \__xeCJK_ulem_skip_punct_end:
8188 {
8189   \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_putbox:
8190   \cs_set_eq:NN \xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_aux:n
8191 }
8192 \cs_new_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
8193 \cs_new_protected:Npn \__xeCJK_ulem_skip_putbox:
8194 {
8195   \tl_if_empty:NF \UL@start
8196   { \box_use_drop:N \UL@box }
8197 }

```

__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。

```

8198 \cs_new_protected:Npn \__xeCJK_ulem_initial:
8199 {
8200   \__xeCJK_ulem_swap_cs:NN
8201   \xeCJK_FullLeft_and_Default: \__xeCJK_ulem_FullLeft_and_Default:
8202   \xeCJK_FullLeft_and_CJK: \__xeCJK_ulem_FullLeft_and_CJK:
8203   \xeCJK_FullLeft_and_Boundary: \__xeCJK_ulem_FullLeft_and_Boundary:
8204   \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
8205   \xeCJK_FullRight_and_CJK: \__xeCJK_ulem_FullRight_and_CJK:
8206   \xeCJK_FullRight_and_CJStarter: \__xeCJK_ulem_FullRight_and_CJStarter:
8207   \xeCJK_FullRight_and_Boundary: \__xeCJK_ulem_FullRight_and_Boundary:
8208   \xeCJK_CJK_and_CJK:N \__xeCJK_ulem_CJK_and_CJK:N
8209   \xeCJK_CJK_and_Boundary:w \__xeCJK_ulem_CJK_and_Boundary:w
8210   \xeCJK@fix@penalty \__xeCJK_ulem_fix_penalty:
8211   \__xeCJK_punct_breakable_kern:n \__xeCJK_ulem_punct_breakable_kern:n
8212   \__xeCJK_Default_and_FullLeft_glue:N \__xeCJK_ulem_Default_and_FullLeft_glue:N
8213   \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N
8214   \__xeCJK_CJK_and_FullLeft_glue:N \__xeCJK_ulem_CJK_and_FullLeft_glue:N
8215   \__xeCJK_CJK_and_FullRight_glue:N \__xeCJK_ulem_CJK_and_FullRight_glue:N

```

```

8216 \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
8217 \q_recursion_tail \q_nil \q_recursion_stop
8218 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
8219 {
8220   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
8221   {
8222     \str_if_eq:nnTF {##1} {####1}
8223     {
8224       \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
8225       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
8226       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
8227       { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
8228     }
8229     {
8230       \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
8231       { \__xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
8232     }
8233   }
8234 }
8235 }
8236 \cs_new_protected:Npn \__xeCJK_ulem_swap_cs:NN #1#2
8237 {
8238   \quark_if_recursion_tail_stop:N #1
8239   \xeCJK_swap_cs:NN #1#2
8240   \__xeCJK_ulem_swap_cs:NN
8241 }

```

\xeCJK_if_ulem_patch:TF 在下划线状态下, ulem 宏包在数学模式或者盒子中使用 \UL@hrest 恢复__ 等的定义, 此时不需要使用 \UL@stop 和 \UL@start 来断开下划线而产生断点。

```

8242 \cs_new:Npn \xeCJK_if_ulem_patch:TF
8243 {
8244   \if_meaning:w \ LA@space
8245   \exp_after:wN \use_ii:nn
8246   \else:
8247     \exp_after:wN \use_i:nn
8248   \fi:
8249 }

```

```

\__xeCJK_ulem_CJK_and_Boundary:w 8250 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_Boundary:w
8251 {
8252   \xeCJK_if_ulem_patch:TF
8253   {
8254     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
8255     { \xeCJK_class_group_end: \CJKe glue }
8256     {
8257       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
8258       { \__xeCJK_ulem_peek_math:w }
8259       { \__xeCJK_ulem_group_end:n { CJK } }
8260     }
8261   }
8262   { \__xeCJK_ulem_CJK_and_Boundary:w }
8263 }
8264 \cs_new_protected:Npn \__xeCJK_ulem_group_end:n #1
8265 {
8266   \xeCJK_class_group_end: \UL@stop
8267   \UL@start { \xeCJK_make_node:n {##1} }
8268   \xeCJK_make_group_tag:
8269   \__xeCJK_boundary_enable_source_space:n {##1}
8270 }

```

__xeCJK_ulem_peek_math:w 用于处理下划线中, 汉字与 \$ 之间有空格的情况¹⁸。

```

8271 \cs_new_protected:Npn \__xeCJK_ulem_peek_math:w
8272 {

```

¹⁸<https://github.com/CTeX-org/ctex-kit/issues/614>

```

8273 \cs_set_eq:NN \__xeCJK_ulem_start:w \__xeCJK_ulem_exp_stop:w
8274 \exp_after:wN \peek_after:Nw
8275 \exp_after:wN \__xeCJK_ulem_peek_math_branches:w
8276 \exp:w \exp_end_continue_f:w
8277 }
8278 \cs_new_protected:Npn \__xeCJK_ulem_peek_math_branches:w
8279 {
8280 \token_if_math_toggle:NTF \l_peek_token
8281 { \xeCJK_class_group_end: \CJKe glue }
8282 { \__xeCJK_ulem_group_end:n { CJK-space } }
8283 }
8284 \cs_new_protected:Npn \__xeCJK_ulem_exp_stop:w
8285 {
8286 \cs_if_eq:NNTF \UL@start \@empty
8287 { \exp_after:wN \exp_stop_f: }
8288 { \exp_after:wN \UL@start }
8289 }

```

```

\__xeCJK_ulem_fix_penalty: 8290 \cs_new_protected:Npn \__xeCJK_ulem_fix_penalty:
8291 {
8292 \xeCJK_if_ulem_patch:TF
8293 { \fix@penalty }
8294 { \__xeCJK_ulem_fix_penalty: }
8295 }

```

在 \xeCJK_class_group_end: 前保存字体状态, 在 __xeCJK_ulem_class_group_begin: 后恢复, 使 \bfseries、\sffamily 等字体声明能跨越下划线内的 CJK 分组边界。

```

8296 \tl_new:N \g__xeCJK_ulem_saved_series_tl
8297 \tl_new:N \g__xeCJK_ulem_saved_shape_tl
8298 \tl_new:N \g__xeCJK_ulem_saved_CJK_family_tl
8299 \tl_new:N \g__xeCJK_ulem_saved_CJK_fam_tl
8300 \cs_new_protected:Npn \__xeCJK_ulem_save_font_state:
8301 {
8302 \tl_gset:Ne \g__xeCJK_ulem_saved_series_tl { \f@series }
8303 \tl_gset:Ne \g__xeCJK_ulem_saved_shape_tl { \f@shape }
8304 \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_family_tl \l_xeCJK_family_tl
8305 \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_fam_tl \CJK@family
8306 }
8307 \cs_new_protected:Npn \__xeCJK_ulem_restore_font_state:
8308 {
8309 \tl_set:Ne \f@series { \g__xeCJK_ulem_saved_series_tl }
8310 \tl_set:Ne \f@shape { \g__xeCJK_ulem_saved_shape_tl }
8311 \tl_set_eq:NN \l_xeCJK_family_tl \g__xeCJK_ulem_saved_CJK_family_tl
8312 \tl_set_eq:NN \CJK@family \g__xeCJK_ulem_saved_CJK_fam_tl
8313 }

```

```

8314 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_CJK:N
8315 {
8316 \xeCJK_if_ulem_patch:TF
8317 {
8318 \__xeCJK_ulem_save_font_state:
8319 \xeCJK_class_group_end:
8320 \UL@stop \__xeCJK_ulem_ccglue: \UL@start
8321 \__xeCJK_ulem_class_group_begin:
8322 \__xeCJK_ulem_restore_font_state:
8323 \xeCJK_select_font:
8324 \xeCJK_fallback_symbol:NN
8325 \CJKsymbol
8326 }
8327 { \__xeCJK_ulem_CJK_and_CJK:N }
8328 }

```

```

8329 \cs_new_protected:Npn \__xeCJK_ulem_class_group_begin:
8330 {
8331 \xeCJK_class_group_begin:
8332 \xeCJK_clear_Boundary_and_CJK_toks:
8333 }

```

```

\__xeCJK_ulem_between_CJK_blocks:nnN 8334 \cs_new_protected:Npn \__xeCJK_ulem_between_CJK_blocks:nnN #1#2
8335 {
8336   \xeCJK_if_ulem_patch:TF
8337   {
8338     \__xeCJK_ulem_save_font_state:
8339     \xeCJK_class_group_end:
8340     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
8341     \xeCJK_class_group_begin:
8342     \xeCJK_clear_Boundary_and_CJK_toks:
8343     \__xeCJK_ulem_restore_font_state:
8344     \__xeCJK_switch_font:nn {#1} {#2}
8345     \xeCJK_fallback_symbol:NN
8346     \CJKsymbol
8347   }
8348   {
8349     \skip_horizontal:N \l__xeCJK_ccglue_skip
8350     \__xeCJK_switch_font:nn {#1} {#2}
8351     \xeCJK_fallback_symbol:NN
8352     \CJKsymbol
8353   }
8354 }

\__xeCJK_ulem_Default_and_FullLeft_glue:N 8355 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
8356 {
8357   \xeCJK_if_ulem_patch:TF
8358   {
8359     \UL@stop
8360     \__xeCJK_ulem_skip_punct_begin:
8361     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8362     \UL@start
8363   }
8364   { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
8365 }

\__xeCJK_ulem_Boundary_and_FullLeft_glue:N 8366 \cs_new_protected:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
8367 {
8368   \xeCJK_if_ulem_patch:TF
8369   {
8370     \UL@stop
8371     \__xeCJK_ulem_skip_punct_begin:
8372     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8373     \UL@start
8374   }
8375   { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
8376 }

\__xeCJK_ulem_CJK_and_FullLeft_glue:N 8377 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
8378 {
8379   \xeCJK_if_ulem_patch:TF
8380   {
8381     \xeCJK_class_group_end:
8382     \UL@stop
8383     \__xeCJK_ulem_skip_punct_begin:
8384     \__xeCJK_ulem_punct_ccglue:
8385     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8386     \UL@start
8387     \__xeCJK_ulem_class_group_begin:
8388     \xeCJK_select_punct_font:
8389   }
8390   { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
8391 }

\__xeCJK_ulem_Default_and_FullRight_glue:N 8392 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
8393 {
8394   \xeCJK_if_ulem_patch:TF
8395   {
8396     \UL@stop
8397     \__xeCJK_ulem_skip_punct_begin:

```

```

8398     \__xeCJK_punct_if_long:NTF #1
8399     { \xeCJK_allow_break: }
8400     { \xeCJK_no_break: }
8401     \__xeCJK_punct_if_middle:NT #1
8402     {
8403         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
8404         \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
8405     }
8406     \UL@start
8407 }
8408 { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
8409 }

```

```

\__xeCJK_ulem_CJK_and_FullRight_glue:N 8410 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
8411 {
8412     \xeCJK_if_ulem_patch:TF
8413     {
8414         \xeCJK_class_group_end:
8415         \UL@stop
8416         \__xeCJK_ulem_skip_punct_begin:
8417         \__xeCJK_punct_if_long:NTF #1
8418         { \xeCJK_allow_break: }
8419         { \xeCJK_no_break: }
8420         \__xeCJK_punct_if_middle:NT #1
8421         {
8422             \__xeCJK_ulem_punct_ccglue:
8423             \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
8424             \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
8425         }
8426         \UL@start
8427         \__xeCJK_ulem_class_group_begin:
8428         \xeCJK_select_punct_font:
8429     }
8430     { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
8431 }

```

```

\__xeCJK_ulem_FullLeft_and_Default: 8432 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_Default:
8433 {
8434     \xeCJK_if_ulem_patch:TF
8435     {
8436         \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
8437         {
8438             \xeCJK_get_punct_bounds:No \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8439             \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8440             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
8441             \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8442         }
8443         { \xeCJK_class_group_end: \UL@stop }
8444         \__xeCJK_ulem_skip_punct_end:
8445         \xeCJK_no_break:
8446         \UL@start
8447     }
8448     { \__xeCJK_ulem_FullLeft_and_Default: }
8449 }

```

```

\__xeCJK_ulem_FullLeft_and_Boundary: 8450 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_Boundary:
8451 {
8452     \xeCJK_if_ulem_patch:TF
8453     {
8454         \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
8455         {
8456             \xeCJK_get_punct_bounds:No \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8457             \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8458             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
8459             \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8460         }
8461         { \xeCJK_class_group_end: \UL@stop }
8462         \__xeCJK_ulem_skip_punct_end:

```

```

8463         \xeCJK_no_break:
8464         \UL@start
8465         \tex_ignorespaces:D
8466     }
8467     { \__xeCJK_ulem_FullLeft_and_Boundary: }
8468 }

\__xeCJK_ulem_FullLeft_and_CJK: 8469 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_CJK:
8470 {
8471     \xeCJK_if_ulem_patch:TF
8472     {
8473         \xeCJK_FullLeft_and_Default:
8474         \__xeCJK_ulem_class_group_begin:
8475         \xeCJK_select_font:
8476     }
8477     { \__xeCJK_ulem_FullLeft_and_CJK: }
8478 }

\__xeCJK_ulem_FullRight_and_Default: 8479 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Default:
8480 {
8481     \xeCJK_if_ulem_patch:TF
8482     {
8483         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8484         \xeCJK_class_group_end:
8485         \UL@stop
8486         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8487         \__xeCJK_ulem_skip_punct_end:
8488         \UL@start
8489     }
8490     { \__xeCJK_ulem_FullRight_and_Default: }
8491 }

\__xeCJK_ulem_FullRight_and_Boundary: 8492 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Boundary:
8493 {
8494     \xeCJK_if_ulem_patch:TF
8495     {
8496         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8497         \xeCJK_class_group_end:
8498         \UL@stop
8499         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8500         \__xeCJK_ulem_skip_punct_end:
8501         \UL@start
8502         \tex_ignorespaces:D
8503     }
8504     { \__xeCJK_ulem_FullRight_and_Boundary: }
8505 }

\__xeCJK_ulem_FullRight_and_CJK: 8506 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJK:
8507 {
8508     \xeCJK_if_ulem_patch:TF
8509     {
8510         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8511         \xeCJK_class_group_end:
8512         \UL@stop
8513         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8514         \__xeCJK_ulem_punct_ccglue:
8515         \__xeCJK_ulem_skip_punct_end:
8516         \UL@start
8517         \__xeCJK_ulem_class_group_begin:
8518         \xeCJK_select_font:
8519     }
8520     { \__xeCJK_ulem_FullRight_and_CJK: }
8521 }

```

__xeCJK_ulem_FullRight_and_CJStarter: 在全角右标点与严格行首禁则字符之间, 先于标点胶插入禁则, 并保持 xeCJKfntef 对标点胶和 CJK 字距的下划线处理。

```
8522 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJStarter:
```

```

8523 {
8524   \xeCJK_if_ulem_patch:TF
8525   {
8526     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8527     \xeCJK_class_group_end:
8528     \UL@stop
8529     \xeCJK_no_break:
8530     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8531     \__xeCJK_ulem_punct_ccglue:
8532     \__xeCJK_ulem_skip_punct_end:
8533     \UL@start
8534     \__xeCJK_ulem_class_group_begin:
8535     \xeCJK_select_font:
8536   }
8537   { \__xeCJK_ulem_FullRight_and_CJStarter: }
8538 }

\__xeCJK_ulem_punct_hskip:n 8539 \cs_new_protected:Npn \__xeCJK_ulem_punct_hskip:n
8540 {
8541   \xeCJK_if_ulem_patch:TF
8542   { \xeCJK_ulem_hskip:n }
8543   { \__xeCJK_ulem_punct_hskip:n }
8544 }

\__xeCJK_ulem_punct_breakable_kern:n 8545 \cs_new_protected:Npn \__xeCJK_ulem_punct_breakable_kern:n #1
8546 {
8547   \xeCJK_if_ulem_patch:TF
8548   {
8549     \xeCJK_class_group_end:
8550     \UL@stop \xeCJK_ulem_hskip:n {#1} \UL@start
8551     \__xeCJK_ulem_class_group_begin:
8552     \xeCJK_select_punct_font:
8553   }
8554   { \__xeCJK_ulem_punct_breakable_kern:n {#1} }
8555 }

\__xeCJK_ulem_glue:n 8556 \cs_new_protected:Npn \__xeCJK_ulem_glue:n #1
\__xeCJK_ulem_ccglue:n 8557 {
\__xeCJK_ulem_punct_ccglue:n 8558   \xeCJK_if_ulem_patch:TF
8559   {
8560     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
8561     { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
8562     {
8563       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
8564       { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
8565       { \skip_horizontal:n {#1} }
8566     }
8567   }
8568   { \skip_horizontal:n {#1} }
8569 }
8570 \cs_new_protected:Npn \xeCJK_make_group_tag:
8571 { \tl_set:Nx \l__xeCJK_group_tag_tl { \c__xeCJK_group_tag_tl } }
8572 \tl_new:N \l__xeCJK_group_tag_tl
8573 \tl_const:Nn \c__xeCJK_group_tag_tl
8574 {
8575   T \int_use:N \tex_currentgrouptype:D
8576   L \int_use:N \tex_currentgrouplevel:D
8577 }
8578 \cs_new_protected:Npn \__xeCJK_ulem_ccglue:
8579 { { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders } }
8580 \cs_new_protected:Npn \__xeCJK_ulem_punct_ccglue:
8581 { \__xeCJK_punct_hskip:n { \l__xeCJK_ccglue_skip } }

\__xeCJK_ulem_stream_begin: 线型命令的正文参数由 ulem 自己扫描,不能使用普通 cmd after hook。ulem 会把正文拆进固
\__xeCJK_boundary_ulem_math_body:n 定宽度盒子; framework 算出的左侧 glue 因此通过 \__xeCJK_boundary_use_ulem_glue:n
\__xeCJK_ulem_group_begin:
\__xeCJK_ulem_group_end:
\__xeCJK_ulem_on:n

```

转交 ulem 的 \UL@stop/\UL@start 外层通道, 保留弹性和断行位置且不把间距画上线。本包的线型命令在装饰符号测量前启动 stream; 原生 \uline 等命令和 \xeCJKfntefon 则由共有的 \ULon 补上同一入口。局部布尔避免本包命令在到达 \ULon 时重复启动。原生 ulem 命令嵌套本包线型命令时, 内层会走 ulem 的 \UL@onin 路径而不会另行调用 _xeCJK_ulem_end; 因此 group 入口与 \ULon 必须共用同一个“仅最外层启动”辅助函数。所有嵌套层复用外层 stream, 最外层唯一的 _xeCJK_ulem_end: 把真实末尾 marker 移到外层列表后结束并清除布尔。 \UL@on/\UL@onin 在把正文交给 ulem 扫描前检查其单参数正文是否以字面数学切换符开始; 独立符号命令在 _xeCJK_under_symbol_auxii:nnnnnn 的对应两端使用同一 begin/end 和公式入口。

```

8582 \cs_new_protected:Npn \_xeCJK_boundary_use_ulem_glue:n #1
8583 {
8584   \xeCJK_if_ulem_patch:TF
8585   {
8586     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
8587     { \UL@stop \skip_horizontal:n {#1} \UL@start }
8588     {
8589       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
8590       { \UL@stop \skip_horizontal:n {#1} \UL@start }
8591       { \skip_horizontal:n {#1} }
8592     }
8593   }
8594   { \skip_horizontal:n {#1} }
8595 }
8596 \cs_new_protected:Npn \_xeCJK_ulem_stream_begin:
8597 {
8598   \bool_if:NF \l__xeCJK_ulem_stream_started_bool
8599   {
8600     \bool_set_true:N \l__xeCJK_ulem_stream_started_bool
8601     \_xeCJK_boundary_inline_stream_begin:nn { auto } { stream-ulem }
8602   }
8603 }
8604 \cs_new_protected:Npn \xeCJK_ulem_group_begin:
8605 {
8606   \mode_leave_vertical:
8607   \_xeCJK_ulem_stream_begin:
8608   \c_group_begin_token
8609 }
8610 \cs_new_protected:Npn \xeCJK_ulem_group_end:
8611 { \c_group_end_token }
8612 \cs_new_protected:Npn \xeCJK_ulem_on:n
8613 { \ULon }
8614 \cs_new_eq:NN \_xeCJK_orig_ulem_on: \ULon
8615 \cs_set_protected:Npn \ULon
8616 {
8617   \_xeCJK_ulem_stream_begin:
8618   \_xeCJK_orig_ulem_on:
8619 }
8620 \bool_new:N \l__xeCJK_ulem_stream_started_bool
8621 % ulem 会把源码空格改造成 leaders; 等到参数末尾再检查时, 公式节点已经被
8622 % 装饰节点遮住。若正文语法确实是“公式 $\quad$ 尾随源码空格”, 先让 ulem 排完
8623 % 去掉尾随空格的正文, 在公式节点仍可见时确认公式末尾, 随后把一枚空格
8624 % 交还 ulem 排版; capture 结束时会将它标记为冻结空格。未知宏若消费公式,
8625 % 实际末节点检查仍会失败。
8626 \cs_new_protected:Npn \_xeCJK_boundary_ulem_math_body:n #1
8627 {
8628   \_xeCJK_boundary_if_math_tail:nTF {#1}
8629   {
8630     \_xeCJK_boundary_if_math_tail_space:nTF {#1}
8631     {
8632       \tl_set:Ne \l__xeCJK_boundary_math_tl
8633       { \tl_trim_right_spaces:n {#1} }
8634       \tl_use:N \l__xeCJK_boundary_math_tl
8635       \_xeCJK_boundary_if_last_math_node:T

```

```

8636         { \_xeCJK_boundary_math_end_confirm_space_visible: }
8637         \c_space_tl
8638     }
8639     { #1 \_xeCJK_boundary_math_end:n {#1} }
8640 }
8641 {#1}
8642 }
8643 \cs_new_eq:NN \_xeCJK_ulem_on:n \UL@on
8644 \cs_set_protected:Npn \UL@on #1
8645 {
8646     \_xeCJK_boundary_math_begin:n {#1}
8647     \_xeCJK_ulem_on:n
8648     {
8649         \xeCJK_ulem_left:
8650         \_xeCJK_boundary_ulem_math_body:n {#1}
8651         \xeCJK_ulem_right:
8652     }
8653 }
8654 \cs_new_eq:NN \_xeCJK_orig_ulem_onin:n \UL@onin
8655 \cs_set_protected:Npn \UL@onin #1
8656 {
8657     \_xeCJK_boundary_math_begin:n {#1}
8658     \_xeCJK_orig_ulem_onin:n { \_xeCJK_boundary_ulem_math_body:n {#1} }
8659 }

```

\xeCJKfntefon 扩展 \ULon 的参数。

```

8660 \NewDocumentCommand \xeCJKfntefon { s t- s o }
8661 {
8662     \mode_leave_vertical:
8663     \xeCJK_ulem_boot:NNNn #1#2#3 {#4}
8664     \xeCJK_ulem_on:n
8665 }

```

\CJKKunderline

```

8666 \NewDocumentCommand \CJKKunderline { s t- s o }
8667 {
8668     \xeCJK_ulem_group_begin:
8669     \xeCJK_fntef_boot:nnNNNn { underline } { uline } #1#2#3 {#4}
8670     \cs_set_eq:NN \ULleaders \tex_cleaders:D
8671     \xeCJK_fntef_initial:nnn
8672     { \l__xeCJK_uline_depth_tl }
8673     { \l__xeCJK_uline_sep_tl }
8674     {
8675         \l__xeCJK_uline_format_tl
8676         \tex_vrule:D
8677         height \dim_eval:n { \l__xeCJK_uline_thickness_tl }
8678         depth \c_zero_dim
8679         width .2em
8680     }
8681     \xeCJK_ulem_on:n
8682 }
8683 \NewDocumentCommand \varCJKKunderline { }
8684 { \CJKKunderline - }

```

\CJKKunderwave

```

8685 \NewDocumentCommand \CJKKunderwave { s t- s o }
8686 {
8687     \xeCJK_ulem_group_begin:
8688     \xeCJK_fntef_boot:nnNNNn { underwave } { uwave } #1#2#3 {#4}
8689     \cs_set_eq:NN \ULleaders \tex_xleaders:D
8690     \xeCJK_fntef_initial:nnn
8691     { \l__xeCJK_uwave_depth_tl }
8692     { \l__xeCJK_uwave_sep_tl }
8693     { \l__xeCJK_uwave_format_tl \l__xeCJK_uwave_symbol_tl }
8694     \xeCJK_ulem_on:n
8695 }

```

\CJKKunderdblline

```

8696 \NewDocumentCommand \CJKKunderdblline { s t- s o }
8697 {

```

```

8698 \xeCJK_ulem_group_begin:
8699 \xeCJK_fntef_boot:nnNNNn { underdblline } { udbline } #1#2#3 {#4}
8700 \cs_set_eq:NN \ULleaders \tex_cleaders:D
8701 \xeCJK_fntef_initial:nnn
8702 { \l__xeCJK_udbline_depth_tl }
8703 { \l__xeCJK_udbline_sep_tl }
8704 {
8705 \l__xeCJK_udbline_format_tl
8706 \vbox_top:n
8707 {
8708 \tex_hrulerule:D
8709 height \dim_eval:n { \l__xeCJK_udbline_thickness_tl }
8710 depth \c_zero_dim
8711 width .2em
8712 \tex_kern:D \dim_eval:n { \l__xeCJK_udbline_gap_tl }
8713 \tex_hrulerule:D
8714 height \dim_eval:n { \l__xeCJK_udbline_thickness_tl }
8715 depth \c_zero_dim
8716 width .2em
8717 }
8718 }
8719 \xeCJK_ulem_on:n
8720 }

\CJKsout 8721 \NewDocumentCommand \CJKsout { s t- s o }
8722 {
8723 \xeCJK_ulem_group_begin:
8724 \xeCJK_fntef_boot:nnNNNn { sout } { sout } #1#2#3 {#4}
8725 \cs_set_eq:NN \ULleaders \tex_cleaders:D
8726 \xeCJK_fntef_initial:nn
8727 {
8728 \l__xeCJK_sout_format_tl
8729 \tex_vrulerule:D
8730 height \dim_eval:n { \l__xeCJK_sout_thickness_tl }
8731 depth \c_zero_dim
8732 width .2em
8733 }
8734 {
8735 \box_move_up:nn
8736 { \l__xeCJK_sout_height_tl - \box_ht:N \l__xeCJK_fntef_box / 2 }
8737 { \box_use:N \l__xeCJK_fntef_box }
8738 }
8739 \xeCJK_ulem_on:n
8740 }

\CJKxout 8741 \NewDocumentCommand \CJKxout { s t- s o }
8742 {
8743 \xeCJK_ulem_group_begin:
8744 \xeCJK_fntef_boot:nnNNNn { xout } { xout } #1#2#3 {#4}
8745 \cs_set_eq:NN \ULleaders \tex_cleaders:D
8746 \xeCJK_fntef_initial:nn
8747 {
8748 \l__xeCJK_xout_format_tl
8749 \tex_kern:D -.1 em $/$
8750 \tex_kern:D -.1 em
8751 }
8752 {
8753 \box_move_up:nn
8754 { \box_dp:N \l__xeCJK_fntef_box / 2 }
8755 { \box_use:N \l__xeCJK_fntef_box }
8756 }
8757 \xeCJK_ulem_on:n
8758 }

\CJKunderanyline 8759 \NewDocumentCommand \CJKunderanyline { s t- s o m m }
8760 {
8761 \xeCJK_ulem_group_begin:
8762 \xeCJK_ulem_boot:NNNn #1#2#3 {#4}

```

```

8763 \cs_set_eq:NN \ULleaders \tex_cleaders:D
8764 \xeCJK_fntef_initial:nn
8765 {#6}
8766 {
8767   \box_move_down:nn
8768   {#5}
8769   { \box_use:N \l__xeCJK_fntef_box }
8770 }
8771 \tl_if_empty:NF \l__xeCJK_ulem_boxdepth_tl
8772 { \box_set_dp:Nn \ULC@box { \l__xeCJK_ulem_boxdepth_tl } }
8773 \tl_if_empty:NF \l__xeCJK_ulem_sep_tl
8774 {
8775   \bool_set_true:N \l__xeCJK_fntef_bool
8776   \dim_set:Nn \l__xeCJK_fntef_dim
8777   { \l__xeCJK_ulem_sep_tl + \box_dp:N \ULC@box }
8778 }
8779 \xeCJK_ulem_on:n
8780 }

```

\xeCJK_fntef_boot:nnNNn 处理参数问题。

```

8781 \cs_new_protected:Npn \xeCJK_fntef_boot:nnNNn #1#2#3#4#5#6
8782 {
8783   \bool_lazy_or:nnT {#3} {#5}
8784   { \bool_set_false:c { l__xeCJK_#2_skip_bool } }
8785   \bool_if:NT #4
8786   { \bool_set_true:c { l__xeCJK_#2_subtract_bool } }
8787   \tl_if_novalue:nF {#6}
8788   { \keys_set:nn { xeCJK / options / #1 } {#6} }
8789   \bool_set_eq:Nc \l__xeCJK_ulem_skip_bool { l__xeCJK_#2_skip_bool }
8790   \bool_set_eq:Nc \l__xeCJK_ulem_hidden_bool { l__xeCJK_#2_hidden_bool }
8791   \bool_set_eq:Nc \l__xeCJK_ulem_subtract_bool { l__xeCJK_#2_subtract_bool }
8792   \tl_set_eq:Nc \l__xeCJK_ulem_text_format_tl { l__xeCJK_#2_text_format_tl }
8793 }
8794 \cs_new_protected:Npn \xeCJK_ulem_boot:NNNn #1#2#3#4
8795 {
8796   \bool_lazy_or:nnT {#1} {#3}
8797   { \bool_set_false:N \l__xeCJK_ulem_skip_bool }
8798   \bool_if:NT #2
8799   { \bool_set_true:N \l__xeCJK_ulem_subtract_bool }
8800   \tl_if_novalue:nF {#4}
8801   { \keys_set:nn { xeCJK / options / ulem } {#4} }
8802 }

```

\xeCJK_fntef_initial:n 不支持下划线的嵌套使用。下划线嵌套使用时,里层的下划线会被放在盒子里,不能折行。

```

8803 \cs_new_protected:Npn \xeCJK_fntef_initial:n
8804 {
8805   \bool_if:NTF \l__xeCJK_nest_bool
8806   { \__xeCJK_warning:n { fntef-nesting } }
8807   {
8808     \bool_set_true:N \l__xeCJK_nest_bool
8809     \__xeCJK_restore_shipout_fntef:
8810   }
8811   \xeCJK_fntef_sbox:n
8812 }
8813 \cs_new_protected:Npn \xeCJK_fntef_initial:nn #1
8814 {
8815   \xeCJK_fntef_initial:n {#1}
8816   \bool_if:NF \l__xeCJK_fntef_bool
8817   { \dim_zero:N \l__xeCJK_fntef_dim }
8818   \markoverwith
8819 }
8820 \cs_new_protected:Npn \xeCJK_fntef_initial:nnn #1#2#3
8821 {
8822   \xeCJK_fntef_initial:n {#3}
8823   \bool_if:NF \l__xeCJK_fntef_bool
8824   {

```

```

8825         \bool_set_true:N \l__xeCJK_fntef_bool
8826         \dim_set:Nn \l__xeCJK_fntef_dim {#1}
8827     }
8828     \markoverwith
8829     {
8830         \box_move_down:nn
8831         { \l__xeCJK_fntef_dim + \box_ht:N \l__xeCJK_fntef_box }
8832         { \box_use:N \l__xeCJK_fntef_box }
8833     }
8834     \dim_set:Nn \l__xeCJK_fntef_dim { #2 + \box_dp:N \ULC@box }
8835 }
8836 \box_new:N \l__xeCJK_fntef_box
8837 \cs_new_eq:NN \xeCJKfntefbox \l__xeCJK_fntef_box
8838 \bool_new:N \l__xeCJK_nest_bool
8839 \bool_new:N \l__xeCJK_fntef_bool
8840 \__xeCJK_msg_new:nn { fntef-nesting }
8841 { Nesting~is~not~supported. }

```

`\l__xeCJK_fntef_dim` 记录下划线或者下划符号的深度,以便它们嵌套使用时能自动调整好距离。`\ULdepth` 被 `ulem` 初始化为 `\maxdimen`。下划线嵌套时,`ulem` 要使用它作计算,可能会溢出。为简便起见,`\l__xeCJK_fntef_dim` 与 `\ULdepth` 共用一个寄存器。

```

8842 \cs_new_eq:NN \l__xeCJK_fntef_dim \ULdepth

```

`\xeCJK_fntef_sbox:n` 与 `\hcoffin_set:Nn` 和 $\text{\LaTeX 2}_{\epsilon}$ 的 `\sbox` 功能类似,确保颜色的正确。虽然 `coffin` 可以更方便的操作盒子,但速度要慢一点。并且,我们的需求也比较简单,就不用它了。

```

8843 \cs_new_protected:Npn \xeCJK_fntef_sbox:n #1
8844 {
8845     \__xeCJK_boundary_capture_suspend:
8846     \hbox_set:Nn \l__xeCJK_fntef_box
8847     { \color_ensure_current: #1 }
8848     \__xeCJK_boundary_capture_resume:
8849 }

```

最合适的是用 `xtemplate` 宏包来实现,但是比较难于用 `\xeCJKsetup` 来统一设置,所以这里还是用土办法。

```

8850 \keys_define:nn { xeCJK / options }
8851 {
8852     underdot / symbol          .tl_set:N = \l__xeCJK_udot_symbol_tl ,
8853     underdot / depth          .tl_set:N = \l__xeCJK_udot_depth_tl ,
8854     underdot / sep            .tl_set:N = \l__xeCJK_udot_sep_tl ,
8855     underdot / format         .tl_set:N = \l__xeCJK_udot_format_tl ,
8856     underdot / textformat     .tl_set:N = \l__xeCJK_udot_text_format_tl ,
8857     underdot / boxdepth       .tl_set:N = \l__xeCJK_udot_boxdepth_tl ,
8858     symbol / sep              .tl_set:N = \l__xeCJK_symbol_sep_tl ,
8859     symbol / boxdepth         .tl_set:N = \l__xeCJK_symbol_boxdepth_tl ,
8860     symbol / textformat       .tl_set:N = \l__xeCJK_symbol_text_format_tl ,
8861     underline / skip          .bool_set:N = \l__xeCJK_uline_skip_bool ,
8862     underline / hidden        .bool_set:N = \l__xeCJK_uline_hidden_bool ,
8863     underline / subtract      .bool_set:N = \l__xeCJK_uline_subtract_bool ,
8864     underline / thickness     .tl_set:N = \l__xeCJK_uline_thickness_tl ,
8865     underline / depth         .tl_set:N = \l__xeCJK_uline_depth_tl ,
8866     underline / sep           .tl_set:N = \l__xeCJK_uline_sep_tl ,
8867     underline / format        .tl_set:N = \l__xeCJK_uline_format_tl ,
8868     underline / textformat    .tl_set:N = \l__xeCJK_uline_text_format_tl ,
8869     underdblline / skip       .bool_set:N = \l__xeCJK_udblline_skip_bool ,
8870     underdblline / hidden     .bool_set:N = \l__xeCJK_udblline_hidden_bool ,
8871     underdblline / subtract    .bool_set:N = \l__xeCJK_udblline_subtract_bool ,
8872     underdblline / thickness   .tl_set:N = \l__xeCJK_udblline_thickness_tl ,
8873     underdblline / depth      .tl_set:N = \l__xeCJK_udblline_depth_tl ,
8874     underdblline / sep        .tl_set:N = \l__xeCJK_udblline_sep_tl ,
8875     underdblline / format     .tl_set:N = \l__xeCJK_udblline_format_tl ,
8876     underdblline / textformat .tl_set:N = \l__xeCJK_udblline_text_format_tl ,
8877     underdblline / gap        .tl_set:N = \l__xeCJK_udblline_gap_tl ,
8878     underwave / skip          .bool_set:N = \l__xeCJK_uwave_skip_bool ,

```

```

8879   underwave / hidden      .bool_set:N = \l__xeCJK_uwave_hidden_bool ,
8880   underwave / subtract    .bool_set:N = \l__xeCJK_uwave_subtract_bool ,
8881   underwave / symbol      .tl_set:N = \l__xeCJK_uwave_symbol_tl ,
8882   underwave / depth       .tl_set:N = \l__xeCJK_uwave_depth_tl ,
8883   underwave / sep         .tl_set:N = \l__xeCJK_uwave_sep_tl ,
8884   underwave / format      .tl_set:N = \l__xeCJK_uwave_format_tl ,
8885   underwave / textformat .tl_set:N = \l__xeCJK_uwave_text_format_tl ,
8886   sout / skip             .bool_set:N = \l__xeCJK_sout_skip_bool ,
8887   sout / hidden          .bool_set:N = \l__xeCJK_sout_hidden_bool ,
8888   sout / subtract         .bool_set:N = \l__xeCJK_sout_subtract_bool ,
8889   sout / thickness       .tl_set:N = \l__xeCJK_sout_thickness_tl ,
8890   sout / height          .tl_set:N = \l__xeCJK_sout_height_tl ,
8891   sout / format          .tl_set:N = \l__xeCJK_sout_format_tl ,
8892   sout / textformat      .tl_set:N = \l__xeCJK_sout_text_format_tl ,
8893   xout / skip            .bool_set:N = \l__xeCJK_xout_skip_bool ,
8894   xout / hidden          .bool_set:N = \l__xeCJK_xout_hidden_bool ,
8895   xout / subtract        .bool_set:N = \l__xeCJK_xout_subtract_bool ,
8896   xout / format          .tl_set:N = \l__xeCJK_xout_format_tl ,
8897   xout / textformat      .tl_set:N = \l__xeCJK_xout_text_format_tl ,
8898   ulem / skip            .bool_set:N = \l__xeCJK_ulem_skip_bool ,
8899   ulem / hidden          .bool_set:N = \l__xeCJK_ulem_hidden_bool ,
8900   ulem / subtract        .bool_set:N = \l__xeCJK_ulem_subtract_bool ,
8901   ulem / sep             .tl_set:N = \l__xeCJK_ulem_sep_tl ,
8902   ulem / boxdepth        .tl_set:N = \l__xeCJK_ulem_boxdepth_tl ,
8903   ulem / textformat      .tl_set:N = \l__xeCJK_ulem_text_format_tl
8904 }
8905 \clist_map_inline:nn
8906 { underdot , underline , underdblline , underwave , sout , xout , ulem }
8907 {
8908   \keys_define:nn { xeCJK / options }
8909     { #1 .meta:nn = { xeCJK / options / #1 } { ##1 } }
8910 }
8911 \keys_set:nn { xeCJK / options }
8912 {
8913   underdot / symbol      = \normalfont . ,
8914   underdot / depth       = 0.20 em ,
8915   underdot / sep         = 0.04 em ,
8916   symbol / sep           = \c_zero_dim ,
8917   underline / skip       = true ,
8918   underline / thickness  = \ULthickness ,
8919   underline / depth      = 0.20 em ,
8920   underline / sep        = 0.07 em ,
8921   underdblline / skip    = true ,
8922   underdblline / thickness = \ULthickness ,
8923   underdblline / depth   = 0.20 em ,
8924   underdblline / sep     = 0.17 em ,
8925   underdblline / gap     = 1.1 pt ,
8926   underwave / skip       = true ,
8927   underwave / symbol     = \sixly \tex_char:D 58 \exp_stop_f: ,
8928   underwave / depth      = 0.20 em ,
8929   underwave / sep        = 0.00 em ,
8930   sout / skip            = true ,
8931   sout / thickness       = \ULthickness ,
8932   sout / height          = 0.35 em ,
8933   xout / skip            = true
8934 }

```

```

\CJKunderanysymbol 8935 \NewDocumentCommand \CJKunderanysymbol { o m m m }
8936 {
8937   \xeCJK_under_symbol:nnnnn { symbol } { symbol } {#1} {#2} {#3} {#4}
8938 }

```

`\CJKunderdot` 是 `\CJKunderanysymbol` 的特殊情况。`CJKfntef` 原来使用的是数学符号 `\cdot`，这里改成更合适的 `.`。

```

8939 \NewDocumentCommand \CJKunderdot { o m }
8940 {

```

```

8941 \xeCJK_under_symbol:nnnnnn { underdot } { udot }
8942 { #1 }
8943 { \l__xeCJK_udot_depth_tl }
8944 { \l__xeCJK_udot_format_tl \l__xeCJK_udot_symbol_tl }
8945 { #2 }
8946 }

```

\xeCJK_under_symbol:nnnnnn 当处在下划线中时,我们先断开下划线,在分组外设置下划符号。

```

8947 \cs_new_protected:Npn \xeCJK_under_symbol:nnnnnn
8948 {
8949   \xeCJK_if_ulem_patch:TF
8950   { \__xeCJK_under_symbol_auxi:nnnnnn }
8951   { \__xeCJK_under_symbol_auxii:nnnnnn }
8952 }
8953 \cs_new_protected:Npn \__xeCJK_under_symbol_auxi:nnnnnn #1#2#3#4#5#6
8954 {
8955   \xeCJK_ulem_right: \UL@stop
8956   \group_begin:
8957     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
8958     \use:c { l__xeCJK_#2_text_format_tl }
8959     \UL@start \xeCJK_ulem_right_node:
8960     #6
8961     \xeCJK_ulem_right: \UL@stop
8962   \group_end:
8963   \UL@start \xeCJK_ulem_right_node:
8964 }
8965 \cs_new_protected:Npn \__xeCJK_under_symbol_auxii:nnnnnn #1#2#3#4#5#6
8966 {
8967   \__xeCJK_boundary_inline_stream_begin:nn { auto } { stream-ulem }
8968   \__xeCJK_boundary_math_begin:n {#6}
8969   \group_begin:
8970     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
8971     \__xeCJK_under_symbol_text_format:c { l__xeCJK_#2_text_format_tl }
8972     #6
8973     \__xeCJK_boundary_math_end:n {#6}
8974     \xeCJK_ulem_right:
8975   \group_end:
8976   \xeCJK_ulem_right_node:
8977   \__xeCJK_boundary_inline_stream_end:n { auto }
8978 }
8979 \cs_new_protected:Npn \xeCJK_under_symbol_initial:nnnnn #1#2#3#4#5
8980 {
8981   \tl_if_novalue:nF {#3}
8982   { \keys_set:nn { xeCJK / options / #1 } {#3} }
8983   \xeCJK_fntef_sbox:n {#5}
8984   \bool_if:NTF \l__xeCJK_fntef_bool
8985   { \xeCJK_make_under_symbol:n { \l__xeCJK_fntef_dim } }
8986   {
8987     \bool_set_true:N \l__xeCJK_fntef_bool
8988     \xeCJK_make_under_symbol:n {#4}
8989   }
8990   \tl_if_empty:cF { l__xeCJK_#2_boxdepth_tl }
8991   {
8992     \box_set_dp:Nn \l__xeCJK_under_symbol_box
8993     { \use:c { l__xeCJK_#2_boxdepth_tl } }
8994   }
8995   \dim_set:Nn \l__xeCJK_fntef_dim
8996   { \use:c { l__xeCJK_#2_sep_tl } + \box_dp:N \l__xeCJK_under_symbol_box }
8997   \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
8998   \__xeCJK_restore_shipout_CJKsymbol:
8999 }
9000 \cs_new_protected:Npn \__xeCJK_under_symbol_text_format:N #1
9001 {
9002   \tl_if_empty:NF #1
9003   { \xeCJK_ulem_right: #1 \xeCJK_ulem_right_node: }
9004 }
9005 \cs_generate_variant:Nn \__xeCJK_under_symbol_text_format:N { c }

```

```
9006 \box_new:N \l__xeCJK_under_symbol_box
```

\xeCJK_make_under_symbol:n 我们量取“一”的宽度作为汉字的宽度。

```
9007 \cs_new_protected:Npn \xeCJK_make_under_symbol:n #1
9008 {
9009   \hbox_set:Nn \l__xeCJK_under_symbol_box
9010   {
9011     \box_move_down:nn { #1 + \box_ht:N \l__xeCJK_fntef_box }
9012     {
9013       \hbox_to_zero:n
9014       {
9015         \xeCJK_select_font:
9016         \tex_kern:D \tex_fontcharwd:D \tex_font:D "4E00 \exp_stop_f:
9017         \tex_hss:D \box_use:N \l__xeCJK_fntef_box \tex_hss:D
9018       }
9019     }
9020   }
9021 }
```

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

```
9022 \cs_new_protected:Npn \_xeCJK_restore_shipout_CJKsymbol:
9023 {
9024   \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
9025   { \xeCJK_swap_cs:NN \CJKsymbol \_xeCJK_under_CJKsymbol:N }
9026   \_xeCJK_restore_shipout_fntef:
9027   \xeCJK_cs_clear:N \_xeCJK_restore_shipout_CJKsymbol:
9028 }
9029 \cs_new_protected:Npn \_xeCJK_restore_shipout_fntef:
9030 {
9031   \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
9032   {
9033     \bool_set_false:N \l__xeCJK_fntef_bool
9034     \dim_zero:N \l__xeCJK_fntef_dim
9035   }
9036   \xeCJK_cs_clear:N \_xeCJK_restore_shipout_fntef:
9037 }
9038 \tl_new:N \l__xeCJK_fntef_shipout_tl
9039 \xeCJK_add_to_shipout:n { \l__xeCJK_fntef_shipout_tl }
```

_xeCJK_under_CJKsymbol:N 盒子放在汉字的左侧, 比较容易处理状态转移的问题。

```
9040 \cs_new_protected:Npn \_xeCJK_under_CJKsymbol:N
9041 {
9042   \box_use:N \l__xeCJK_under_symbol_box
9043   \xeCJK_no_break: \_xeCJK_under_CJKsymbol:N
9044 }
```

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

```
9045 \NewDocumentEnvironment { CJKfilltwosides } { 0 { c } m }
9046 {
9047   \use:e { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
9048   \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
9049 }
9050 {
9051   \endminipage
9052   \ignorespacesafterend
9053 }
9054 \NewDocumentEnvironment { CJKfilltwosides* } { 0 { c } m +b }
9055 {
9056   \mode_leave_vertical:
9057   \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
9058   \tl_set:Nn \arraystretch { 1 }
```

```

9059 \cs_if_free:NF \extrarowheight
9060 { \cs_set_eq:NN \extrarowheight \c_zero_dim }
9061 \use:e { \__xeCJK_fill_two_sides:nnn {#1} { \dim_eval:n {#2} } } {#3}
9062 }
9063 { \ignorespacesafterend }
9064 \cs_new_protected:Npn \__xeCJK_fill_two_sides:nnn #1#2#3
9065 {
9066   \dim_compare:nNnTF {#2} > \c_zero_dim
9067   {
9068     \hbox_set:Nn \l__xeCJK_tmp_box
9069     { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
9070     \dim_compare:nNnTF {#2} > { \box_wd:N \l__xeCJK_tmp_box }
9071     { \tabular [#1] { @ { } p {#2} @ { } } #3 \endtabular }
9072     { \box_use:N \l__xeCJK_tmp_box }
9073   }
9074   { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
9075 }

```

`\xeCJK_fntef_hfilll:` `colortbl` 将表格 `c` 列用于填充的 `\hfil` 改为了更高阶的 `fill`, 影响到了 `CJKfillltwosides*`。因此, 我们也要用高阶的 `filll`。

```

9076 \cs_new_protected:Npn \xeCJK_fntef_hfilll:
9077 { \skip_horizontal:N \c__xeCJK_filll_skip }
9078 \skip_const:Nn \c__xeCJK_filll_skip { \c_zero_dim plus 1 filll }
9079 </fntef>

```

5.20 xeCJK-listings

仿照 `luatexja` 宏包中 `lltp-listings` 的处理, 支持 `listings` 宏包。

```

9080 <*listings>
9081 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
9082 \ProcessOptions \scan_stop:
9083 \RequirePackage { xeCJK }
9084 \RequirePackage { listings }
9085 \lst@AddToHook { Init } { \__xeCJK_listings_initial_hook: }
9086 \lst@AddToHook { SelectCharTable } { \__xeCJK_listings_toks_hook: }
9087 \lst@AddToHook { OutputBox }
9088 {
9089   \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
9090   \l__xeCJK_restore_listings_toks_tl
9091   \__xeCJK_listings_output_CM:
9092 }
9093 \lst@AddToHook { PreSet } { \bool_set_true:N \l__xeCJK_listings_env_bool }

```

`\__xeCJK_listings_initial_hook:` 为使代码行号结果正确, 需要在 `\lst@numberstyle` 中恢复 `\XeTeXinterchartoks`。在 `listings` 环境中换页时, 对 `\XeTeXinterchartoks` 的修改会影响到页眉和页脚, 需要在 `\shipout` 盒子中恢复成正常定义。加入 `\tex_noindent:D` 是为了进入水平模式, 防止汉字出现在首行的时候可能会产生额外空行。`\lst@prebreak` 和 `\lst@postbreak` 是在 `\discretionary` 中直接输出的, 应该恢复正常的 `\XeTeXinterchartoks`。

```

9094 \cs_new_protected:Npn \__xeCJK_listings_initial_hook:
9095 {
9096   \tex_noindent:D
9097   \bool_gset_false:N \g__xeCJK_listings_CM_bool
9098   \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
9099   \xeCJK_add_to_shipout:n { \l__xeCJK_restore_listings_toks_tl }
9100   \lst@ifbreaklines
9101   \cs_set_eq:NN \__xeCJK_listings_CJK_toks_hook: \__xeCJK_listings_breaklines_toks:
9102   \tl_if_empty:NF \lst@prebreak
9103   { \tl_put_left:Nn \lst@prebreak { \l__xeCJK_restore_listings_toks_tl } }
9104   \tl_if_empty:NF \lst@postbreak
9105   { \tl_put_left:Nn \lst@postbreak { \l__xeCJK_restore_listings_toks_tl } }

```

```

9106 \fi:
9107 \int_set:Nn \l__xeCJK_listings_max_char_int
9108 { \lst@ifec 255 \else: 127 \fi: }
9109 }
9110 \int_new:N \l__xeCJK_listings_max_char_int

```

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

```

9111 \cs_new_protected:Npn \__xeCJK_listings_toks_hook:
9112 {
9113   \tl_clear:N \l__xeCJK_restore_listings_toks_tl
9114   \seq_map_function:NN
9115     \g__xeCJK_class_seq \__xeCJK_backup_inter_class_toks:n
9116   \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
9117   {
9118     \str_if_eq:nnF { ##1 } { Boundary }
9119     {
9120       \xeCJK_inter_class_toks:nnn { Boundary } { ##1 }
9121       { \__xeCJK_listings_process_Default:n { ##1 } }
9122     }
9123   }
9124   \xeCJK_inter_class_toks:nnn { Boundary } { CM }
9125   { \__xeCJK_listings_process_CM:n { 0 } }
9126   \__xeCJK_listings_CJK_toks_hook:
9127 }

```

`\__xeCJK_backup_inter_class_toks:n` 注意, 给 `\XeTeXinterchartoks` 赋空值, 会导致 XeTeX 崩溃!

```

9128 \cs_new_protected:Npn \__xeCJK_backup_inter_class_toks:n #1
9129 {
9130   \tl_set:Ne \l__xeCJK_tmp_tl
9131   { \xeCJK_get_inter_class_toks:nn { Boundary } {#1} }
9132   \tl_put_right:Ne \l__xeCJK_restore_listings_toks_tl
9133   {
9134     \xeCJK_inter_class_toks:nnn { Boundary } {#1}
9135     {
9136       \tl_if_empty:NTF \l__xeCJK_tmp_tl
9137       { \exp_not:N \prg_do_nothing: }
9138       { \exp_not:o \l__xeCJK_tmp_tl }
9139     }
9140   }
9141 }
9142 \tl_new:N \l__xeCJK_restore_listings_toks_tl

```

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

```

\__xeCJK_listings_breaklines_toks:
9143 \cs_new_protected:Npn \__xeCJK_listings_CJK_toks_hook:
9144 {
9145   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
9146   { \__xeCJK_listings_process_CJK:n { 2 } }
9147   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
9148   { \__xeCJK_listings_process_CJK:n { 2 } }
9149   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
9150   { \__xeCJK_listings_process_CJK:n { 2 } }
9151   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
9152   { \__xeCJK_listings_process_CJK:n { 2 } }
9153   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
9154   { \__xeCJK_listings_process_CJK:n { 2 } }
9155   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
9156   { \__xeCJK_listings_process_CM:n { 0 } }
9157   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
9158   { \__xeCJK_listings_process_CM:n { 0 } }
9159   \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
9160   { \__xeCJK_listings_process_CJK:n { 2 } }
9161   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
9162   {
9163     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }

```

```

9164         { \_xeCJK_listings_process_CJK:nN { 2 } }
9165     }
9166 }
9167 \cs_new_protected:Npn \_xeCJK_listings_breaklines_toks:
9168 {
9169     \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
9170     { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9171     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
9172     { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9173     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
9174     { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9175     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
9176     { \_xeCJK_listings_process_CM:nN { 0 } }
9177     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
9178     { \_xeCJK_listings_process_CM:nN { 0 } }
9179     \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
9180     { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9181     \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
9182     { \_xeCJK_listings_process_FullLeft:nN { 2 } }
9183     \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
9184     { \_xeCJK_listings_process_FullRight:nN { 2 } }
9185     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
9186     {
9187         \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
9188         { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9189     }
9190 }

```

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

```

9191 \cs_new_protected:Npn \_xeCJK_listings_process_Default:nN #1#2
9192 {
9193     \int_compare:nNnTF
9194     { \xeCJK_token_value_charcode:N #2 } > \l__xeCJK_listings_max_char_int
9195     {
9196         \token_if_letter:NTF #2
9197         { \lst@ProcessLetter #2 }
9198         { \lst@ProcessOther #2 }
9199     }
9200     { \_xeCJK_listings_output_Default:nN {#1} #2 }
9201 }

```

输出时, 要注意把对应的 \XeTeXinterchartoks 清空掉, 否则会造成死循环。 \scan_stop: 是造边界, 输出 \group_end:。

```

9202 \cs_new_protected:Npn \_xeCJK_listings_output_Default:nN #1#2
9203 {
9204     \group_begin:
9205     \xeCJK_clear_inter_class_toks:nn { Boundary } {#1}
9206     \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \group_end: }
9207     #2
9208     \scan_stop:
9209 }

```

_xeCJK_listings_process_CJK:nN 对 CJK 字符类的处理。

```

9210 \cs_new_protected:Npn \_xeCJK_listings_process_CJK:nN #1#2
9211 {
9212     \token_if_letter:NTF #2
9213     { \_xeCJK_listings_process_letter:nN {#1} #2 }
9214     { \_xeCJK_listings_process_other:nN {#1} #2 }
9215 }

```

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

```

9216 \cs_new_protected:Npn \_xeCJK_listings_append:nN #1#2

```

```

9217 {
9218   \int_add:Nn \lst@length { #1 - 1 }
9219   \lst@Append #2
9220 }

```

在 letter 类中区分汉字和西文字母。

```

9221 \cs_new_protected:Npn \__xeCJK_listings_process_letter:nN
9222 {
9223   \lst@whitespacefalse
9224   \bool_if:NTF \l__xeCJK_listings_letter_bool
9225     { \lst@lettertrue }
9226     {
9227       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9228       \bool_set_true:N \l__xeCJK_listings_letter_bool
9229     }
9230   \__xeCJK_listings_append:nN
9231 }
9232 \bool_new:N \l__xeCJK_listings_letter_bool
9233 \cs_new_protected:Npn \__xeCJK_listings_process_other:nN #1#2
9234 {
9235   \lst@whitespacefalse
9236   \bool_if:NTF \l__xeCJK_listings_letter_bool
9237     {
9238       \lst@Output \lst@letterfalse
9239       \bool_set_false:N \l__xeCJK_listings_letter_bool
9240     }
9241     { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
9242   \cs_set_eq:NN \lst@lastother #2
9243   \__xeCJK_listings_append:nN {#1} #2
9244 }

```

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

```

9245 \cs_new_protected:Npn \__xeCJK_listings_process_breaklines_CJK:nN
9246 {
9247   \lst@whitespacefalse
9248   \bool_if:NTF \l__xeCJK_listings_letter_bool
9249     {
9250       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2 { \lst@Output }
9251       \lst@lettertrue
9252     }
9253     {
9254       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9255       \bool_set_true:N \l__xeCJK_listings_letter_bool
9256     }
9257   \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one_int
9258   \__xeCJK_listings_append:nN
9259 }
9260 \cs_new_protected:Npn \__xeCJK_listings_process_FullLeft:nN #1#2
9261 {
9262   \lst@whitespacefalse
9263   \bool_if:NTF \l__xeCJK_listings_letter_bool
9264     {
9265       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2
9266         {
9267           \int_compare:nNnTF \l__xeCJK_listings_flag_int = 3
9268             { \bool_if:NT \l__xeCJK_punct_breakable_bool { \lst@Output } }
9269             { \lst@Output }
9270         }
9271       \lst@lettertrue
9272     }
9273     {
9274       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9275       \bool_set_true:N \l__xeCJK_listings_letter_bool
9276     }

```

```

9277 \int_set:Nn \l__xeCJK_listings_flag_int { 2 }
9278 \__xeCJK_listings_append:nN {#1} #2
9279 }
9280 \cs_new_protected:Npn \__xeCJK_listings_process_FullRight:nN #1#2
9281 {
9282   \lst@whitespacefalse
9283   \bool_if:NTF \l__xeCJK_listings_letter_bool
9284   {
9285     \int_compare:nNt \l__xeCJK_listings_flag_int < 2
9286     { \__xeCJK_punct_if_long:NT #2 { \lst@Output } }
9287     \lst@lettertrue
9288   }
9289   {
9290     \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9291     \bool_set_true:N \l__xeCJK_listings_letter_bool
9292   }
9293   \int_set:Nn \l__xeCJK_listings_flag_int { 3 }
9294   \__xeCJK_listings_append:nN {#1} #2
9295 }
9296 \int_new:N \l__xeCJK_listings_flag_int
\lst@AppendLetter 9297 \cs_set_protected:Npn \lst@AppendLetter
\lst@AppendOther 9298 {
9299   \bool_if:NTF \l__xeCJK_listings_letter_bool
9300   {
9301     \lst@Output \lst@lettertrue
9302     \bool_set_false:N \l__xeCJK_listings_letter_bool
9303   }
9304   { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
9305   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
9306   \lst@Append
9307 }
9308 \cs_set_protected:Npn \lst@AppendOther
9309 {
9310   \bool_if:NTF \l__xeCJK_listings_letter_bool
9311   {
9312     \lst@Output \lst@letterfalse
9313     \bool_set_false:N \l__xeCJK_listings_letter_bool
9314   }
9315   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
9316   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
9317   \tex_futurelet:D \lst@lastother \lst@Append
9318 }

```

__xeCJK_listings_process_CM:nN CM类作为 letter 处理, 不用增加 \lst@length。

```

9319 \cs_new_protected:Npn \__xeCJK_listings_process_CM:nN
9320 {
9321   \reverse_if:N \lst@ifflexible
9322   \bool_gset_true:N \g__xeCJK_listings_CM_bool
9323   \fi:
9324   \__xeCJK_listings_process_letter:nN
9325 }

```

__xeCJK_listings_output_CM: 在使用 columns=fixed 选项时, listings 会在输出盒子中的每个字符之间加入 \hss, 这就破坏了 X_YTeX 将基本字和组合标识正确的组合起来。

```

9326 \cs_new_protected:Npn \__xeCJK_listings_output_CM:
9327 {
9328   \reverse_if:N \lst@ifflexible
9329   \bool_if:NT \g__xeCJK_listings_CM_bool
9330   {
9331     \bool_gset_false:N \g__xeCJK_listings_CM_bool
9332     \xeCJK_cs_clear:N \lst@FillOutputBox
9333     \cs_set_eq:NN \CJKglue \tex_hss:D
9334   }
9335   \fi:
9336 }

```

```
9337 \bool_new:N \g__xeCJK_listings_CM_bool
```

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

```
9338 \cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
9339 {
9340   \token_if_active:NTF #3
9341     { #1#3 }
9342     {
9343       \token_if_cs:NTF #3
9344         { #2#3 }
9345         {
9346           \int_compare:nNnTF { `#3 } > { \l__xeCJK_listings_max_char_int }
9347             { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
9348             { #2#3 }
9349         }
9350     }
9351 }
9352 \cs_set_eq:NN \lst@ifnextcharactive \__xeCJK_listings_peek_active_loop:TF
```

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

当 \lstinline 出现在宏参数中时, TeX 参数传递将 catcode 6 的 parameter token (通常是 #) 双写为成对的相邻 token。 \tl_set_rescan:Nnn 底层的 \scantokens 在字符串化阶段会再次将 catcode 6 token 双写, 导致输出中数量翻倍。修复方法是在 rescan 前逐 token 扫描 \l__xeCJK_tmp_tl, 将每个 catcode 6 的 token 替换为保留原字符码的 active token (catcode 13), 这是 listings 期望的字符类型。最初的实现用 \regex_replace_all:nnN 把所有 catcode 6 token 硬替换成 U+0023, 对一般 # 没问题, 但当用户用 \catcode 把别的字符 (如 &) 改为 parameter 时, 原字符码会被丢失, 导致输出错字 (#879)。

```
9353 \cs_new_protected:Npn \__xeCJK_listings_rescan:Nn #1#2
9354 {
9355   \__xeCJK_listings_set_escape:
9356   \tl_set:Nn \l__xeCJK_tmp_tl {#2}
9357   \__xeCJK_listings_escape_backslash:
9358   \tl_clear:N \l__xeCJK_tmpb_tl
9359   \tl_map_inline:Nn \l__xeCJK_tmp_tl
9360     {
9361       \token_if_parameter:NTF ##1
9362       {
9363         \tl_put_right:Ne \l__xeCJK_tmpb_tl
9364         {
9365           \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:N
9366           \char_generate:nn { \int_value:w `##1 } { 13 }
9367         }
9368       }
9369       { \tl_put_right:Nn \l__xeCJK_tmpb_tl { ##1 } }
9370     }
9371   \tl_set_eq:NN \l__xeCJK_tmp_tl \l__xeCJK_tmpb_tl
9372   \tl_set_rescan:Nno #1
9373     { \char_set_catcode_active:n { 35 } }
9374     { \l__xeCJK_tmp_tl }
9375 }
9376 \cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
9377 {
9378   \__xeCJK_listings_rescan:Nn \l__xeCJK_tmp_tl {#1}
9379   \tl_put_right:No \lst@arg { \l__xeCJK_tmp_tl }
9380 }
9381 \cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
9382 \cs_new_protected:Npn \__xeCJK_listings_inline_group:w
9383 {
```

```

9384 \exp_after:wN \__xeCJK_listings_inline_group:n
9385 \exp_after:wN { \if_int_compare:w `} = \c_zero_int \fi:
9386 }
9387 \cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
9388 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
9389 {
9390 \__xeCJK_listings_rescan:Nn \lst@arg {#1}
9391 \lst@InlineGJEnd
9392 }

```

__xeCJK_listings_boundary_stream_end: \lstinline 自己只声明可选参数,正文由多个内部扫描器读取,普通 cmd after hook 会过早结束。分隔符路径经过已经打开外层组、即将读取分隔符的 \lstinline@, 花括号路径则绕过它而直接进入 \lst@InlineG; 两个入口都启动 capture 并设置局部标志。 \lst@DeInit 返回到该外层组后用 \aftergroup 安排单 token 结束函数。分隔符、活动字符、花括号与 short-inline 路径都调用 \lst@DeInit, 因此共用同一结束适配点。

```

9393 \clist_map_inline:nn { \lstinline , \lstinline@ , \lst@InlineG , \lst@DeInit }
9394 { \__xeCJK_boundary_reserve:n {#1} }
9395 \bool_new:N \l__xeCJK_listings_boundary_bool
9396 \cs_new_protected:Npn \__xeCJK_listings_boundary_stream_end:
9397 { \__xeCJK_boundary_inline_stream_end:n { auto } }
9398 \cs_new_eq:NN \__xeCJK_orig_listings_inline:n \lstinline@
9399 \cs_set_protected:Npn \lstinline@ #1
9400 {
9401 \__xeCJK_boundary_inline_stream_begin:n { auto }
9402 \bool_set_true:N \l__xeCJK_listings_boundary_bool
9403 \__xeCJK_orig_listings_inline:n #1
9404 }
9405 \cs_new_eq:NN \__xeCJK_orig_listings_inline_group: \lst@InlineG
9406 \cs_set_protected:Npn \lst@InlineG
9407 {
9408 \__xeCJK_boundary_inline_stream_begin:n { auto }
9409 \bool_set_true:N \l__xeCJK_listings_boundary_bool
9410 \__xeCJK_orig_listings_inline_group:
9411 }
9412 \tl_gput_right:Nn \lst@DeInit
9413 {
9414 \bool_if:NT \l__xeCJK_listings_boundary_bool
9415 { \aftergroup \__xeCJK_listings_boundary_stream_end: }
9416 }

```

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

```

9417 \group_begin:
9418 \cs_set:Npn \__xeCJK_tmp:w #1
9419 {
9420 \group_end:
9421 \cs_new_protected:Npn \__xeCJK_listings_set_escape:
9422 { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
9423 \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
9424 { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
9425 }
9426 \use:n
9427 {
9428 \char_set_catcode_active:N \
9429 \__xeCJK_tmp:w
9430 }
9431 { \ }

```

__xeCJK_listings_escape_backslash: \catcode 为 12 的\ 需要双写转义。

```

9432 \cs_new_protected:Npe \__xeCJK_listings_escape_backslash:
9433 {
9434 \tl_replace_all:Nnn \exp_not:N \l__xeCJK_tmp_tl
9435 { \c_backslash_str }

```

```

9436      { \c_backslash_str \c_backslash_str }
9437    }

9438 </listings>

9439 <@@=xunadd>

```

5.21 xunicode-addon

```
9440 <*xunicode>
```

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

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

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

```

9441 \bool_lazy_or:nnF
9442 { \sys_if_engine_xetex_p: }
9443 { \sys_if_engine luatex_p: }
9444 {
9445   \msg_new:nnnn { xunicode-addon } { xetex-luatex }
9446   { This~package~requires~either~XeTeX~or~LuaTeX~to~function.}
9447   {
9448     You~must~change~your~typesetting~engine~to,~e.g.,\\
9449     "xelatex"~or~"lualatex"~instead~of~plain~"latex"~or~"pdflatex".
9450   }
9451   \msg_critical:nn { xunicode-addon } { xetex-luatex }
9452 }
9453 \RequirePackage { xparse }

```

宏包选项是编码的名字。

```

9454 \clist_new:N \g__xunadd_encname_clist
9455 \tl_if_exist:NT \UTFencname
9456 { \clist_gput_right:Ne \g__xunadd_encname_clist { \UTFencname } }
9457 \DeclareOption*
9458 { \clist_gput_right:No \g__xunadd_encname_clist \CurrentOption }
9459 \ProcessOptions \scan_stop:

```

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

```

9460 \@ifpackageloaded { xunicode } { }
9461 {
9462   \clist_get:NNF \g__xunadd_encname_clist \UTFencname
9463   {
9464     \cs_if_exist:NTF \UnicodeEncodingName
9465     { \tl_set:Ne \UTFencname { \UnicodeEncodingName } }
9466     {
9467       \sys_if_engine_xetex:TF
9468       { \tl_set:Nn \UTFencname { EU1 } }
9469       { \tl_set:Nn \UTFencname { EU2 } }
9470     }
9471     \clist_gset_eq:NN \g__xunadd_encname_clist \UTFencname
9472   }
9473   \sys_if_engine_xetex:TF
9474   { \RequirePackage { xunicode } }
9475   {
9476     \cs_set_eq:NN \__xunadd_tmp:w \XeTeXpicfile
9477     \cs_set_eq:NN \XeTeXpicfile \prg_do_nothing:
9478     \RequirePackage { xunicode }
9479     \cs_set_eq:NN \XeTeXpicfile \__xunadd_tmp:w
9480   }
9481 }
9482 \AtEndOfPackage { \__xunadd_reload:N \g__xunadd_encname_clist }

```

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

```

9483 \RenewDocumentCommand \ReloadXunicode { m }
9484 {
9485   \clist_set:Ne \l__xunadd_encname_clist {#1}
9486   \__xunadd_reload:N \l__xunadd_encname_clist
9487 }
9488 \cs_new_protected:Npn \__xunadd_reload:N #1
9489 {
9490   \cs_set_eq:NN \__xunadd_tmp:w \iftipaonetoken
9491   \cs_set_eq:NN \iftipaonetoken \scan_stop:
9492   \use:e
9493   {
9494     \ExplSyntaxOff
9495     \char_set_catcode_letter:n { 64 }
9496     \exp_not:N \clist_map_function:NN \exp_not:N #1 \__xunadd_reload_aux:n
9497     \bool_if:NTF \l__kernel_expl_bool
9498       { \ExplSyntaxOn }
9499       { \ExplSyntaxOff }
9500     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
9501   }
9502   \cs_set_eq:NN \iftipaonetoken \__xunadd_tmp:w
9503 }
9504 \cs_new_protected:Npn \__xunadd_reload_aux:n #1
9505 {
9506   \cs_if_exist:cTF { T@ #1 }
9507   {
9508     \tl_set:Nn \UTFencname {#1}
9509     \clist_if_in:NnF \g__xunadd_encname_clist {#1}
9510       { \clist_gput_right:Nn \g__xunadd_encname_clist {#1} }
9511     \file_input:n { xunicode.sty }
9512     \file_input:n { xunicode-extra.def }
9513   }
9514   { \msg_error:nnn { xunicode-addon } { encoding-unknown } {#1} }
9515 }
9516 \clist_new:N \l__xunadd_encname_clist
9517 \msg_new:nnnn { xunicode-addon } { encoding-unknown }
9518 { Encoding~scheme~"#1"~unknown. }
9519 {
9520   You~may~use \\\
9521   \token_to_str:N \usepackage [ #1 , \encodingdefault ] \{fontenc\} \\\
9522   before~xunicode-addon~or~xunicode.
9523 }

```

`\DeclareUTFmathsymbols` 将文本符号定义为 `\protected` 宏后, 为了与 `hyperref` 的书签功能兼容需要作一点额外处理。

```

9524 \RenewDocumentCommand \DeclareUTFmathsymbols { m }
9525 {
9526   \bool_if:NT \l__xunadd_math_as_UTF_text_bool
9527   {
9528     \seq_map_inline:Nn \l__xunadd_math_as_UTF_text_seq
9529       { \__xunadd_declare_math_as_UTF_text:n {##1} }
9530     \bool_set_false:N \l__xunadd_math_as_UTF_text_bool
9531   }
9532 }
9533 \seq_new:N \l__xunadd_math_as_UTF_text_seq
9534 \seq_set_from_clist:Nn \l__xunadd_math_as_UTF_text_seq
9535 { hbar , Finv , aleph , beth , gimel , daleth , Game }
9536 \bool_new:N \l__xunadd_math_as_UTF_text_bool
9537 \RenewDocumentCommand \UseMathAsText { }
9538 {
9539   \math@s@text@true
9540   \bool_set_true:N \l__xunadd_math_as_UTF_text_bool
9541 }
9542 \@onlypreamble \UseMathAsText
9543 \cs_new_protected:Npn \__xunadd_declare_math_as_UTF_text:n #1
9544 {

```

```

9545 \cs_if_exist:cTF {#1}
9546 {
9547   \cs_new_eq:cc { keepmathUTF #1 } {#1}
9548   \cs_gset_protected:cpe {#1}
9549   {
9550     \exp_not:N \mode_if_math:TF
9551     { \exp_not:c { keepmathUTF #1 } }
9552     { \exp_not:c { text #1 } }
9553   }
9554   \tl_put_right:Ne \l__xunadd_hyperref_hook_tl
9555   { \cs_set_eq:NN \exp_not:c {#1} \exp_not:c { text #1 } }
9556 }
9557 { \cs_new:cpe {#1} { \exp_not:c { text #1 } } }
9558 }
9559 \tl_new:N \l__xunadd_hyperref_hook_tl
9560 \AtBeginDocument
9561 {
9562   \cs_if_free:NF \pdfstringdefDisableCommands
9563   { \pdfstringdefDisableCommands { \l__xunadd_hyperref_hook_tl } }
9564 }

```

`\__xunadd_glyph_if_exist_p:n` 判断字符在当前字体中是否存在。

```

\__xunadd_glyph_if_exist:nTF
9565 \prg_new_conditional:Npnn \__xunadd_glyph_if_exist:n #1 { p , T , F , TF }
9566 {
9567   \tex_iffontchar:D \tex_font:D \tex_numexpr:D #1 \scan_stop:
9568   \prg_return_true: \else: \prg_return_false: \fi:
9569 }

```

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

```

9570 \RenewDocumentCommand \UndeclareUTFcharacter { O { \UTFencname } m m }
9571 {
9572   \__xunadd_if_csname:nTF {#3}
9573   { \UndeclareTextCommand {#3} }
9574   { \exp_args:Nc \UndeclareTextCommand { \tl_to_str:n {#3} } }
9575   {#1}
9576 }

```

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

```

9577 \RenewDocumentCommand \UndeclareUTFcomposite { O { \UTFencname } m m m }
9578 {
9579   \__xunadd_if_csname:nTF {#3}
9580   { \__xunadd_undeclare_composite:Nnnn #3 }
9581   { \exp_args:Nc \__xunadd_undeclare_composite:Nnnn { \tl_to_str:n {#3} } }
9582   {#1} {#4} {#2}
9583 }
9584 \cs_new_protected:Npn \__xunadd_undeclare_composite:Nnnn #1#2#3#4
9585 { \cs_undefine:c { \__xunadd_composite_cs:Nnn #1 {#2} {#3} } }

```

```

\__xunadd_composite_cs:Nnn 9586 \cs_new:Npe \__xunadd_composite_cs:Nnn #1#2#3
\__xunadd_composite_cs:nnn 9587 { \c_backslash_str #2 \exp_not:N \token_to_str:N #1 - \exp_not:N \tl_to_str:n {#3} }
9588 \cs_new:Npe \__xunadd_composite_cs:nnn #1#2#3
9589 { \c_backslash_str #2 #1 - \exp_not:N \tl_to_str:n {#3} }

```

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

```

\DeclareUTFcharacter[\UTFencname]{x0149}{\n}
9590 \prg_new_conditional:Npnn \__xunadd_if_csname:n #1 { TF }
9591 {
9592   \tl_if_single_token:nTF {#1}
9593   {
9594     \token_if_cs:NTF #1
9595     { \prg_return_true: }
9596     {
9597       \token_if_active:NTF #1
9598       { \prg_return_true: }
9599       { \prg_return_false: }

```

```

9600     }
9601   }
9602   { \prg_return_false: }
9603 }

```

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

```

9604 \RenewDocumentCommand \DeclareUTFcharacter { 0 { \UTFencname } m m }
9605 {
9606   \cs_if_exist_use:cF
9607   { __xunadd_restore_ \tl_to_str:n {#3} : }
9608   {
9609     \__xunadd_if_csname:nTF {#3}
9610     { \__xunadd_declare_character:Nnn #3 }
9611     { \__xunadd_declare_character:cnm { \tl_to_str:n {#3} } }
9612     {#1} {#2}
9613   }
9614 }

```

__xunadd_restore_cmd:N 恢复 \hbar 和 \nobreakspace 为原本定义。

```

9615 \cs_new_protected:cpn
9616 { __xunadd_restore_ \tl_to_str:n { \hbar } : }
9617 { \__xunadd_restore_cmd:N \hbar }
9618 \cs_new_protected:cpn
9619 { __xunadd_restore_ \tl_to_str:n { \nobreakspace } : }
9620 { \__xunadd_restore_cmd:N \nobreakspace }
9621 \cs_new_protected:Npn \__xunadd_restore_cmd:N #1
9622 { \__xunadd_restore_cmd:Ne #1 { ? - \token_to_str:N #1 } }
9623 \cs_new_protected:Npn \__xunadd_restore_cmd:Nn #1#2
9624 {
9625   \cs_if_free:cF {#2}
9626   { \__xunadd_restore_cmd:Nc #1 {#2} }
9627 }
9628 \cs_new_protected:Npn \__xunadd_restore_cmd:NN #1#2
9629 {
9630   \cs_gset_eq:NN #1 #2
9631   \cs_undefine:N #2
9632 }
9633 \cs_generate_variant:Nn \__xunadd_restore_cmd:Nn { Ne }
9634 \cs_generate_variant:Nn \__xunadd_restore_cmd:NN { Nc }

```

__xunadd_declare_character:Nnn 通过 \tex_Uchar:D 直接由 Unicode #3 得到编码 #2 下的符号命令 #1 对应的实际字符。
\DeclareUTFSymbol 的参数格式与 \DeclareTextSymbol 完全一致。

```

9635 \cs_new_protected:Npn \__xunadd_declare_character:Nnn #1#2#3
9636 {
9637   \__xunadd_provide_text_command_default:N #1
9638   \exp_after:wN \__xunadd_declare_character:NNen
9639   \tex_Uchar:D \__xunadd_check_slot:n {#3} \exp_stop_f:
9640   #1 { \token_to_str:N #1 } {#2}
9641 }
9642 \cs_generate_variant:Nn \__xunadd_declare_character:Nnn { c }

```

\DeclareUTFSymbol \DeclareUTFCommand 只能用于定义不带参数的符号命令。

```

\DeclareUTFCommand
9643 \NewDocumentCommand \DeclareUTFSymbol { m 0 { \UTFencname } m }
9644 { \__xunadd_declare_character:Nnn #1 {#2} {#3} }
9645 \NewDocumentCommand \DeclareUTFCommand { m 0 { \UTFencname } m }
9646 { \__xunadd_text_command:Nnnn #1 { \token_to_str:N #1 } {#2} {#3} }
9647 \cs_new_protected:Npn \__xunadd_text_command:Nnnn #1#2#3#4
9648 { \DeclareTextCommand #1 {#3} { \__xunadd_text_command:nn {#2} {#4} } }
9649 \cs_generate_variant:Nn \__xunadd_text_command:Nnnn { No }
9650 \cs_new_protected:Npn \__xunadd_text_command:nn #1#2
9651 {
9652   \__xunadd_begin_hook:nn {#1} {#2}
9653   #2
9654   \__xunadd_end_hook:nn {#1} {#2}
9655 }

```

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

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

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

```
9656 \cs_new_protected:Npn \_xunadd_provide_text_command_default:N #1
9657 {
9658   \cs_if_exist:cF { ? \token_to_str:N #1 }
9659   {
9660     \cs_if_free:cF { ? - \token_to_str:N #1 }
9661     {
9662       \exp_args:NNv \ProvideTextCommandDefault #1
9663       { ? - \token_to_str:N #1 }
9664     }
9665   }
9666 }
```

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

```
9667 \cs_new_protected:Npn \_xunadd_declare_character:NNnn #1#2#3#4
9668 { \DeclareTextCommand #2 {#4} { \_xunadd_text_character:nN {#3} {#1} } }
9669 \cs_new_protected:Npn \_xunadd_text_character:nN #1#2
9670 {
9671   \_xunadd_begin_hook:nn {#1} {#2}
9672   \_xunadd_glyph_if_exist:nTF { `#2 }
9673   {#2} { \cs_if_exist_use:cF { ? #1 } {#2} }
9674   \_xunadd_end_hook:nn {#1} {#2}
9675 }
9676 \cs_generate_variant:Nn \_xunadd_declare_character:NNnn { NNe }
```

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

```
9677 \cs_new:Npn \_xunadd_check_slot:n #1
9678 {
9679   \int_eval:n
9680   {
9681     \tl_if_head_eq_charcode:nNTF {#1} x
9682     { " \use_none:n #1 } {#1}
9683   }
9684 }
```

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

```
9685 \RenewDocumentCommand \DeclareUTFcomposite { 0 { \UTFencname } m m m }
9686 {
9687   \_xunadd_if_csname:nTF {#3}
9688   { \_xunadd_declare_composite:Nnnn #3 }
9689   { \_xunadd_declare_composite:cnnn { \tl_to_str:n {#3} } }
9690   {#1} {#4} {#2}
9691 }
```

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

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

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

```

9692 \cs_new_protected:Npn \__xunadd_declare_composite:Nnnn #1#2#3#4
9693 {
9694   \tex_afterassignment:D \use_none_delimit_by_q_stop:w
9695   \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
9696   { \__xunadd_check_slot:n {#4} }
9697   \q_stop
9698 }
9699 \cs_new_protected:Npn \__xunadd_chardef:Nn #1#2
9700 { \tex_chardef:D #1 = \tex_numexpr:D #2 \scan_stop: }
9701 \cs_generate_variant:Nn \__xunadd_chardef:Nn { c }
9702 \cs_generate_variant:Nn \__xunadd_declare_composite:Nnnn { c }

```

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

```

9703 \NewDocumentCommand \DeclareUTFCompositeCommand { m O { \UTFencname } m m }
9704 { \cs_set_protected:cpn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} } {#4} }

```

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

```

9705 \NewDocumentCommand \DeclareUTFCompositeSymbol { m O { \UTFencname } m m }
9706 {
9707   \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
9708   { \__xunadd_check_slot:n {#4} }
9709 }

```

\DeclareUTFComposite 将 #1 设置为编码 #2 下的带一个参数的复合符号命令。

```

9710 \NewDocumentCommand \DeclareUTFComposite { m O { \UTFencname } }
9711 { \use:e { \__xunadd_declare_composite:Nnn \exp_not:N #1 { \token_to_str:N #1 } {#2} } }

```

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

```

9712 \NewDocumentCommand \DeclareUTFEncodedAccent { m O { \UTFencname } m m }
9713 { \__xunadd_declare_encoded:NNnnn \__xunadd_combine_accent:nnNNn #1 {#2} {#3} {#4} }

```

\DeclareUTFEncodedAccents #1 是重音命令, #2 是编码, #3 和 #4 都是组合重音符号的 Unicode。输出 #1 与 #3、#4 的组合。

```

9714 \NewDocumentCommand \DeclareUTFEncodedAccents { m O { \UTFencname } m m }
9715 { \__xunadd_declare_encoded:NNnnn \__xunadd_combine_accents:nnNNn #1 {#2} {#3} {#4} }

```

\DeclareUTFEncodedSymbol #1 是带参数的符号命令, #2 是编码, #3 是组合符号的 Unicode, #4 是基本符号的 Unicode。当 #1 的参数为空时, 输出 #4, 否则是 #1 的参数与 #3 的组合。

```

9716 \NewDocumentCommand \DeclareUTFEncodedSymbol { m O { \UTFencname } m m }
9717 { \__xunadd_declare_encoded:NNnnn \__xunadd_combine_symbol:nnNNn #1 {#2} {#3} {#4} }

```

\DeclareUTFEncodedCircle #1 是带参数的圆圈符号命令, #2 是编码, #3 是组合圆圈符号的 Unicode, #4 是圆圈符号的 Unicode。当 #1 的参数为空时, 输出 #4, 否则是 #1 的参数与 #4 的组合。

```

9718 \NewDocumentCommand \DeclareUTFEncodedCircle { m O { \UTFencname } m m }
9719 { \__xunadd_declare_encoded:NNnnn \__xunadd_combine_circle:nnNNn #1 {#2} {#3} {#4} }

```

\DeclareEncodedCompositeCharacter 9720 \RenewDocumentCommand \DeclareEncodedCompositeCharacter { m m m m }
9721 { \DeclareUTFEncodedSymbol #2 [#1] { "#3 } { "0#4 } }

\DeclareEncodedCompositeAccents 9722 \RenewDocumentCommand \DeclareEncodedCompositeAccents { m m m m }
9723 { \DeclareUTFEncodedAccents #2 [#1] { "#4 } { "#3 } }

\DeclareUTFDoubleEncodedAccent 9724 \NewDocumentCommand \DeclareUTFDoubleEncodedAccent { m O { \UTFencname } m m }
9725 { __xunadd_declare_encoded:NNnnn __xunadd_combine_double_accent:nnNNn #1 {#2} {#3} {#4} }

\DeclareUTFDoubleEncodedSymbol 9726 \NewDocumentCommand \DeclareUTFDoubleEncodedSymbol { m O { \UTFencname } m m }
9727 { __xunadd_declare_encoded:NNnnn __xunadd_combine_double_symbol:nnNNn #1 {#2} {#3} {#4} }

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

```

9728 \cs_new_protected:Npn \_xunadd_declare_composite:Nnn #1#2#3
9729 { \DeclareTextCommand #1 {#3} { \_xunadd_text_composite:nnn {#2} {#3} } }

```

```

\_xunadd_text_composite:nnn 9730 \cs_new_protected:Npn \_xunadd_text_composite:nnn #1#2#3
9731 {
9732   \_xunadd_begin_hook:nn {#1} {#3}
9733   \cs_if_exist:cTF { \_xunadd_composite_cs:nnn {#1} {#2} {#3} }
9734   {
9735     \_xunadd_text_composite:cnn
9736     { \_xunadd_composite_cs:nnn {#1} {#2} {#3} } {#1} {#3}
9737   }
9738   { \cs_if_exist_use:cTF { ? #1 } { {#3} } {#3} }
9739   \_xunadd_end_hook:nn {#1} {#3}
9740 }
9741 \cs_new_protected:Npn \_xunadd_text_composite:Nnn #1#2#3
9742 {
9743   \token_if_chardef:NTF #1
9744   {
9745     \_xunadd_glyph_if_exist:nTF {#1}
9746     {#1} { \cs_if_exist_use:cTF { ? #2 } { {#3} } {#3} }
9747   }
9748   {#1}
9749 }
9750 \cs_generate_variant:Nn \_xunadd_text_composite:Nnn { c }

```

`\_xunadd_declare_encoded:NNnnn` 通过 `\tex_Uchar:D` 直接由重音符号的 Unicode 得到实际字符。

```

9751 \cs_new_protected:Npn \_xunadd_declare_encoded:NNnnn #1#2#3#4#5
9752 {
9753   \exp_after:wN \_xunadd_declare_encoded:NNNNee
9754   \tex_Uchar:D \_xunadd_check_slot:n {#4} \exp_after:wN \exp_stop_f:
9755   \tex_Uchar:D \_xunadd_check_slot:n {#5} \exp_stop_f:
9756   #1 #2 { \token_to_str:N #2 } {#3}
9757 }
9758 \cs_new_protected:Npn \_xunadd_declare_encoded:NNNNnn #1#2#3#4#5#6
9759 { \DeclareTextCommand #4 {#6} { #3 {#5} {#6} {#1} {#2} } }
9760 \cs_generate_variant:Nn \_xunadd_declare_encoded:NNnnn { c }
9761 \cs_generate_variant:Nn \_xunadd_declare_encoded:NNNNnn { NNNNee }

```

`\_xunadd_text_combine:NnnNNn` 若重音命令 #2 与它的参数 #6 的复合已经由 `\DeclareUTFcomposite` 设置, 并且在当前字体中存在该字符, 则直接使用。否则使用组合命令。

```

9762 \cs_new_protected:Npn \_xunadd_text_combine:NnnNNn #1#2#3#4#5#6
9763 {
9764   \_xunadd_begin_hook:nn {#2} {#6}
9765   \cs_if_exist:cTF { \_xunadd_composite_cs:nnn {#2} {#3} {#6} }
9766   {
9767     \_xunadd_text_combine:cNnNNn
9768     { \_xunadd_composite_cs:nnn {#2} {#3} {#6} } #1 {#2} {#4} {#5} {#6}
9769   }
9770   { #1 {#6} {#2} {#4} {#5} }
9771   \_xunadd_end_hook:nn {#2} {#6}
9772 }
9773 \cs_new_protected:Npn \_xunadd_text_combine:NNnNNn #1#2#3#4#5#6
9774 {
9775   \token_if_chardef:NTF #1
9776   { \_xunadd_glyph_if_exist:nTF {#1} {#1} { #2 {#6} {#3} {#4} {#5} } }
9777   {#1}
9778 }
9779 \cs_generate_variant:Nn \_xunadd_text_combine:NNnNNn { c }

```

```

\_xunadd_combine_symbol:nnNNN 9780 \cs_new_protected:Npn \_xunadd_combine_symbol:nnNNN
9781 { \_xunadd_text_combine:NnnNNn \_xunadd_add_symbol:nnNN }
9782 \cs_new_protected:Npn \_xunadd_add_symbol:nnNN #1#2#3#4
9783 {
9784   \tl_if_blank:nTF {#1}
9785   {

```

```

9786     \__xunadd_glyph_if_exist:nTF { `#4 }
9787     {#4}
9788     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
9789   }
9790   {
9791     \__xunadd_glyph_if_exist:nTF { `#3 }
9792     { #1#3 }
9793     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
9794   }
9795 }

```

若组合重音符号的 #3 和基本重音符号 #4 在当前字体中都不存在，则转换到 `\DeclareTextAccentDefault` 设置的编码或者使用 `\DeclareTextCommandDefault` 中设置的命令。0.9999 版以前的 $\text{\XeTeX}$ 需要设置 `\XeTeXinputnormalization` 为 1，才能使用字体中由基础字符和组合符号对应的实际字符；而 0.9999 版以后的 $\text{\XeTeX}$ 默认就启用这个功能，`\XeTeXinputnormalization` 似乎是无效的，怀疑是使用 HarfBuzz 库替代 ICU 进行字体排版的缘故¹⁹。

```

9796 \cs_new_protected:Npn \__xunadd_combine_accent:nnNNn
9797 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accent:nnNN }
9798 \cs_new_protected:Npn \__xunadd_add_accent:nnNN #1#2#3#4
9799 {
9800   \tl_if_blank:nTF {#1}
9801   {
9802     \__xunadd_glyph_if_exist:nTF { `#4 }
9803     {#4}
9804     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
9805   }
9806   {
9807     \__xunadd_glyph_if_exist:nTF { `#3 }
9808     { #1#3 }
9809     {
9810       \__xunadd_glyph_if_exist:nTF { `#4 }
9811       { \add@accent { `#4 } {#1} }
9812       { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
9813     }
9814   }
9815 }

```

```

9816 \cs_new_protected:Npn \__xunadd_combine_accents:nnNNn
9817 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accents:nnNN }
9818 \cs_new_protected:Npn \__xunadd_add_accents:nnNN #1#2#3#4
9819 {
9820   \tl_if_blank:nTF {#1}
9821   { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
9822   {
9823     \__xunadd_glyph_if_exist:nTF { `#3 }
9824     { \__xunadd_glyph_if_exist:nTF { `#4 } }
9825     { \use_ii:nn }
9826     { #1#3#4 }
9827     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3#4 } }
9828   }
9829 }

```

对圆圈中的数字或者字母适当缩小，以适合圆圈的大小。只有字体中存在 U+25EF 时，才使用这里的设置，否则还是 $\text{\LaTeX}$ 中的设置。

```

9830 \cs_new_protected:Npn \__xunadd_combine_circle:nnNNn
9831 { \__xunadd_text_combine:NnnNNn \__xunadd_add_circle:nnNN }
9832 \cs_new_protected:Npn \__xunadd_add_circle:nnNN #1#2#3#4
9833 {
9834   \tl_if_blank:nTF {#1}
9835   {
9836     \__xunadd_glyph_if_exist:nTF { `#4 }

```

¹⁹<http://tug.org/pipermail/xetex/2013-July/024579.html>

```

9837         {#4}
9838         { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
9839     }
9840     {
9841         \__xunadd_glyph_if_exist:nTF { `#4 }
9842         { \__xunadd_add_circle:nN {#1} #4 }
9843         { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
9844     }
9845 }
9846 \cs_new_protected:Npn \__xunadd_add_circle:nN #1#2
9847 {
9848     \hcoffin_set:Nn \l__xunadd_tmp_coffin {#1}
9849     \hcoffin_set:Nn \l__xunadd_circle_coffin {#2}
9850     \fp_set:Nn \l__xunadd_circle_scale_fp
9851     {
9852         \dim_to_decimal_in_unit:nn
9853         {
9854             \fp_use:N \l__xunadd_circle_ratio_fp
9855             \coffin_wd:N \l__xunadd_circle_coffin
9856         }
9857         { \coffin_wd:N \l__xunadd_tmp_coffin }
9858     }
9859     \coffin_scale:Nnn \l__xunadd_tmp_coffin
9860     { \l__xunadd_circle_scale_fp } { \l__xunadd_circle_scale_fp }
9861     \coffin_attach:NnnNnnnn
9862     \l__xunadd_circle_coffin { hc } { vc }
9863     \l__xunadd_tmp_coffin { hc } { vc } { \c_zero_dim } { \c_zero_dim }
9864     \coffin_typeset:Nnnnn \l__xunadd_circle_coffin
9865     { H } { l } { \c_zero_dim } { \c_zero_dim }
9866 }
9867 \fp_new:N \l__xunadd_circle_scale_fp
9868 \coffin_new:N \l__xunadd_tmp_coffin
9869 \coffin_new:N \l__xunadd_circle_coffin

```

\setttextcircledratio 设置圆圈中文字的宽度与圆圈宽度的比例, 预设值为 0.7。

```

9870 \NewDocumentCommand \setttextcircledratio { m }
9871 { \fp_set:Nn \l__xunadd_circle_ratio_fp {#1} }
9872 \fp_new:N \l__xunadd_circle_ratio_fp
9873 \setttextcircledratio { 0.7 }

```

__xunadd_combine_double_accent:nnNn 使 \t 等组合重音符号放在参数的第一个字母的右边。

```

9874 \cs_new_protected:Npn \__xunadd_combine_double_accent:nnNn
9875 { \__xunadd_text_combine:NnnNnn \__xunadd_add_double_accent:nnNN }
9876 \cs_new_protected:Npn \__xunadd_add_double_accent:nnNN #1#2#3#4
9877 {
9878     \tl_if_blank:nTF {#1}
9879     {
9880         \__xunadd_glyph_if_exist:nTF { `#4 }
9881         {#4}
9882         { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
9883     }
9884     {
9885         \__xunadd_glyph_if_exist:nTF { `#3 }
9886         { \__xunadd_add_double_symbol:nN {#1} #3 }
9887         {
9888             \__xunadd_glyph_if_exist:nTF { `#4 }
9889             { \add@accent { `#4 } {#1} }
9890             { \cs_if_exist_use:cTF { ? #2 } { {#1} } { {#1#3} } }
9891         }
9892     }
9893 }

```

__xunadd_combine_double_symbol:nnNn 使 \sliding 等组合重音符号放在参数的第一个字母的右边。

```

9894 \cs_new_protected:Npn \__xunadd_combine_double_symbol:nnNn
9895 { \__xunadd_text_combine:NnnNnn \__xunadd_add_double_symbol:nnNN }

```

```

9896 \cs_new_protected:Npn \__xunadd_add_double_symbol:nnNN #1#2#3#4
9897 {
9898   \tl_if_blank:nTF {#1}
9899   {
9900     \__xunadd_glyph_if_exist:nTF {`#4}
9901     {#4}
9902     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
9903   }
9904   {
9905     \__xunadd_glyph_if_exist:nTF {`#3}
9906     { \__xunadd_add_double_symbol:nN {#1} #3 }
9907     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
9908   }
9909 }

```

__xunadd_add_double_symbol:nN 如果参数的第一个记号是字母类、其他符号类或者由 \chardef 定义, 则将组合符号放在它的右边, 否则不作处理。

```

9910 \cs_new_protected:Npn \__xunadd_add_double_symbol:nN #1#2
9911 {
9912   \tl_if_head_is_N_type:nTF {#1}
9913   {
9914     \exp_after:wN \exp_after:wN \exp_after:wN
9915     \__xunadd_add_double_symbol_aux:NnN \exp_after:wN \exp_after:wN
9916     \tl_head:w #1 \q_stop \exp_after:wN { \use_none:n #1 } #2
9917   }
9918   { #1#2 }
9919 }
9920 \cs_new_protected:Npn \__xunadd_add_double_symbol_aux:NnN #1#2#3
9921 {
9922   \bool_lazy_any:nTF
9923   {
9924     { \token_if_letter_p:N #1 }
9925     { \token_if_other_p:N #1 }
9926     { \token_if_chardef_p:N #1 }
9927   }
9928   { #1#3#2 }
9929   { #1#2#3 }
9930 }

```

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

```

9931 \NewDocumentCommand \AtBeginUTFCommand { s O { } +m }
9932 {
9933   \tl_if_blank:nTF {#2}
9934   {
9935     \IfBooleanTF {#1}
9936     { \xunadd_set_begin_hook:n }
9937     { \xunadd_append_begin_hook:n }
9938   }
9939   { \xunadd_set_begin_hook:nn {#2} }
9940   {#3}
9941 }
9942 \NewDocumentCommand \AtEndUTFCommand { s O { } +m }
9943 {
9944   \tl_if_blank:nTF {#2}
9945   {
9946     \IfBooleanTF {#1}
9947     { \xunadd_set_end_hook:n }
9948     { \xunadd_append_end_hook:n }
9949   }
9950   { \xunadd_set_end_hook:nn {#2} }
9951   {#3}
9952 }

```

```

\xunadd_set_begin_hook:n 9953 \cs_new_protected:Npn \xunadd_set_begin_hook:n
\xunadd_set_end_hook:n

```

```

9954 { \tl_set:Nn \l__xunadd_begin_hook_tl }
9955 \cs_new_protected:Npn \xunadd_append_begin_hook:n
9956 { \tl_put_right:Nn \l__xunadd_begin_hook_tl }
9957 \cs_new_protected:Npn \xunadd_set_end_hook:n
9958 { \tl_set:Nn \l__xunadd_end_hook_tl }
9959 \cs_new_protected:Npn \xunadd_append_end_hook:n
9960 { \tl_put_right:Nn \l__xunadd_end_hook_tl }
9961 \cs_new_protected:Npn \xunadd_set_begin_hook:nn
9962 { \__xunadd_set_cmd_hook:nnn { begin } }
9963 \cs_new_protected:Npn \xunadd_set_end_hook:nn
9964 { \__xunadd_set_cmd_hook:nnn { end } }
9965 \cs_new_protected:Npn \__xunadd_set_cmd_hook:nnn #1#2#3
9966 {
9967   \cs_set_protected:cpn
9968   {
9969     \tl_if_single:nTF {#2}
9970     { \use:c { __xunadd_#1_csname:n } { \token_to_str:N #2 } }
9971     { \__xunadd_set_cmd_hook_aux:Nnwn #2 \q_stop {#1} }
9972   } ##1
9973   {#3}
9974 }
9975 \cs_new:Npn \__xunadd_set_cmd_hook_aux:Nnwn #1#2 \q_stop #3
9976 { \use:c { __xunadd_#3_csname:n } { \token_to_str:N #1 - \tl_to_str:n {#2} } }
9977 \cs_new:Npn \__xunadd_begin_csname:n #1 { __xunadd_begin_#1_hook:n }
9978 \cs_new:Npn \__xunadd_end_csname:n #1 { __xunadd_end_#1_hook:n }
9979 \tl_new:N \l__xunadd_begin_hook_tl
9980 \tl_new:N \l__xunadd_end_hook_tl

```

```

\__xunadd_begin_hook:nn 9981 \cs_new_protected:Npn \__xunadd_begin_hook:nn #1#2
\__xunadd_end_hook:nn 9982 {
9983   \tl_use:N \l__xunadd_begin_hook_tl
9984   \cs_if_exist_use:cF { \__xunadd_begin_csname:n { #1 - \tl_to_str:n {#2} } }
9985   { \cs_if_exist_use:cF { \__xunadd_begin_csname:n {#1} } { \use_none:n } }
9986   {#2}
9987 }
9988 \cs_new_protected:Npn \__xunadd_end_hook:nn #1#2
9989 {
9990   \cs_if_exist_use:cF { \__xunadd_end_csname:n { #1 - \tl_to_str:n {#2} } }
9991   { \cs_if_exist_use:cF { \__xunadd_end_csname:n {#1} } { \use_none:n } }
9992   {#2}
9993   \tl_use:N \l__xunadd_end_hook_tl
9994 }
\DeclareUTF8TIPACCommand 9995 \NewDocumentCommand \DeclareUTF8TIPACCommand { 0 { \UTFencname } m }
9996 { \use:e { __xunadd_text_tipa_command:Nnn \exp_not:N #2 { \token_to_str:N #2 } {#1} } }
9997 \cs_new_protected:Npn \__xunadd_text_tipa_command:Nnn #1#2#3
9998 {
9999   \cs_set_eq:cc { UTF/#3#2 } { #3#2 }
10000   \DeclareTextCommand #1 {#3} { \__xunadd_text_tipa_command:nnn {#3} {#2} }
10001 }
10002 \cs_new_protected:Npn \__xunadd_text_tipa_command:nnn #1#2#3
10003 {
10004   \exp_args:Ncc \__xunadd_check_for_tipa:NNn
10005   { \use_none:n #2 } { UTF/#1#2 } {#3}
10006 }
10007 \cs_new_protected:Npn \__xunadd_check_for_tipa:NNn #1#2#3
10008 {
10009   \tl_if_head_eq_meaning:nNTF {#3} \textipa
10010   {
10011     \exp_after:wN \tipacatchonechar \exp_after:wN
10012     { \exp_after:wN #1 \use_none:n #3 }
10013   }
10014   { #2 {#3} }
10015 }

```

\xunadd_get_slot:nn #1 是编码, #2 是诸如 \textendash 或 \v C 等形式的文本命令, 取得他们对应的字符编码。

```

10016 \cs_new_protected:Npn \xunadd_get_slot:nn #1#2

```

```

10017 { \__xunadd_get_slot:wn #2 \q_nil \q_stop {#1} }
10018 \cs_new_protected:Npn \__xunadd_get_slot:wn #1#2#3 \q_stop #4
10019 {
10020   \int_set:Nn \l_xunadd_slot_int { -1 }
10021   \bool_set_false:N \l_xunadd_rest_bool
10022   \group_begin: \exp_args:Nccc \group_end:
10023   { \__xunadd_get_slot:NNnn }
10024   { #4 \token_to_str:N #1 }
10025   { \__xunadd_composite_cs:Nnn #1 {#4} {#2} }
10026   {#2}
10027   {#3}
10028 }
10029 \int_new:N \l_xunadd_slot_int
10030 \bool_new:N \l_xunadd_rest_bool
10031 \cs_new_protected:Npn \__xunadd_get_slot:NNnn #1#2#3#4
10032 {
10033   \cs_if_free:NF #1
10034   {
10035     \cs_if_exist:NTF #2
10036     { \__xunadd_get_composite_slot:Nn #2 {#4} }
10037     { \__xunadd_get_character_slot:Nn #1 { #3 #4 } }
10038   }
10039 }
10040 \cs_new_protected:Npn \__xunadd_get_composite_slot:Nn #1#2
10041 {
10042   \token_if_chardef:NT #1
10043   {
10044     \int_set:Nn \l_xunadd_slot_int {#1}
10045     \quark_if_nil:nF {#2}
10046     { \bool_set_true:N \l_xunadd_rest_bool }
10047   }
10048 }
10049 \cs_new_protected:Npn \__xunadd_get_character_slot:Nn #1
10050 {
10051   \exp_after:wn \__xunadd_get_character_slot_aux:wn #1
10052   \__xunadd_text_character:nN \q_nil \q_nil \q_stop
10053 }
10054 \cs_new_protected:Npn \__xunadd_get_character_slot_aux:wn
10055 #1 \__xunadd_text_character:nN #2#3#4 \q_stop #5
10056 {
10057   \quark_if_nil:nF {#2}
10058   {
10059     \int_set:Nn \l_xunadd_slot_int { `#3 }
10060     \quark_if_nil:nF {#5}
10061     { \bool_set_true:N \l_xunadd_rest_bool }
10062   }
10063 }

```

\xunadd@microtype@is@charx **microtype** 宏包中使用的函数,我们通过对 \MT@is@charx 打补丁来实现功能。

```

10064 \cs_new_protected_nopar:Npn \xunadd@microtype@is@charx #1 \relax
10065 {
10066   \use:e
10067   { \xunadd_get_slot:nn { \MT@encoding } { \tex_the:D \MT@toks } }
10068   \int_compare:nNnTF \l_xunadd_slot_int < \c_zero_int
10069   { \xunadd@original@is@charx #1 \relax }
10070   {
10071     \cs_set_nopar:Npe \MT@char@ { \int_use:N \l_xunadd_slot_int }
10072     \bool_if:NT \l_xunadd_rest_bool { \MT@noresetfalse }
10073   }
10074 }
10075 \cs_new_protected_nopar:Npn \xunadd@microtype@hook
10076 {
10077   \cs_if_free:NF \MT@is@charx
10078   {
10079     \cs_new_eq:NN \xunadd@original@is@charx \MT@is@charx
10080     \cs_set_eq:NN \MT@is@charx \xunadd@microtype@is@charx

```

```

10081      \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
10082    }
10083  }
10084  \@ifpackageloaded { microtype }
10085  { \use:n } { \AtBeginDocument }
10086  { \xunadd@microtype@hook }

10087 </xunicode>
10088 <*xunextra>

```

我们补充定义 HYPHENATION POINT 和 TWO-EM DASH, 他们默认被归入 CJK 标点符号。

```

10089 \DeclareUTFSymbol\textthyphenationpoint{"2027}
10090 \DeclareUTFSymbol\texttwoemdash{"2E3A}

```

以下内容选自 `xunicode`, 并做了适当修改。

```

10091 \DeclareUTFComposite\textsuperscript
10092 \DeclareUTFComposite\textsubscript
10093 \DeclareUTFEncodedAccent\textsbleftarrow{"20EE}{ "02FF}
10094 \DeclareUTFEncodedAccent\`{"0300}{ "02CB}
10095 \DeclareUTFEncodedAccent\capitalgrave{"0300}{ "02CB}
10096 \DeclareUTFEncodedAccent\'{"0301}{ "02CA}
10097 \DeclareUTFEncodedAccent\capitalacute{"0301}{ "02CA}
10098 \DeclareUTFEncodedAccent\~{"0302}{ "02C6}
10099 \DeclareUTFEncodedAccent\capitalcircumflex{"0302}{ "02C6}
10100 \DeclareUTFEncodedAccent\~{"0303}{ "02DC}
10101 \DeclareUTFEncodedAccent\capitaltilde{"0303}{ "02DC}
10102 \DeclareUTFEncodedAccent\={"0304}{ "02C9}
10103 \DeclareUTFEncodedAccent\capitalmacron{"0304}{ "02C9}
10104 \DeclareUTFEncodedAccent\textoverline{"0305}{ "203E}
10105 \DeclareUTFEncodedAccent\u{"0306}{ "02D8}
10106 \DeclareUTFEncodedAccent\capitalbreve{"0306}{ "02D8}
10107 \DeclareUTFEncodedAccent\{"0307}{ "02D9}
10108 \DeclareUTFEncodedAccent\capitaldotaccent{"0307}{ "02D9}
10109 \DeclareUTFEncodedAccent\'{"0308}{ "00A8}
10110 \DeclareUTFEncodedAccent\capitaldieresis{"0308}{ "00A8}
10111 \DeclareUTFEncodedAccent\m{"0309}{ "0309}
10112 \DeclareUTFEncodedAccent\texthookabove{"0309}{ "0309}
10113 \DeclareUTFEncodedAccent\r{"030A}{ "02DA}
10114 \DeclareUTFEncodedAccent\capitalring{"030A}{ "02DA}
10115 \DeclareUTFEncodedAccent\H{"030B}{ "02DD}
10116 \DeclareUTFEncodedAccent\capitalhungarumlaut{"030B}{ "02DD}
10117 \DeclareUTFEncodedAccent\v{"030C}{ "02C7}
10118 \DeclareUTFEncodedAccent\capitalcaron{"030C}{ "02C7}
10119 \DeclareUTFEncodedAccent\textvbaraccent{"030D}{ "02C8}
10120 \DeclareUTFEncodedAccent\textdoublevbaraccent{"030E}{ "0022}
10121 \DeclareUTFEncodedAccent\U{"030E}{ "0022}
10122 \DeclareUTFEncodedAccent\textdoublegrave{"030F}{ "02F5}
10123 \DeclareUTFEncodedAccent\G{"030F}{ "02F5}
10124 \DeclareUTFEncodedAccent\textdotbreve{"0310}{ "0310}
10125 \DeclareUTFEncodedAccent\textroundcap{"0311}{ "0311}
10126 \DeclareUTFEncodedAccent\newtie{"0311}{ "0311}
10127 \DeclareUTFEncodedAccent\capitalnewtie{"0311}{ "0311}
10128 \DeclareUTFEncodedAccent\textturncommaabove{"0312}{ "02BB}
10129 \DeclareUTFEncodedAccent\textcommaabove{"0313}{ "02BC}
10130 \DeclareUTFEncodedAccent\textrevcommaabove{"0314}{ "02BD}
10131 \DeclareUTFEncodedAccent\overbridge{"0346}{ "0346}
10132 \DeclareUTFEncodedAccent\crtilde{"034A}{ "034A}
10133 \DeclareUTFEncodedAccent\dottedtilde{"034B}{ "034B}
10134 \DeclareUTFEncodedAccent\doubletilde{"034C}{ "034C}
10135 \DeclareUTFEncodedAccent\textrightarrowhead{"0350}{ "02C3}
10136 \DeclareUTFEncodedAccent\textlefthalfring{"0351}{ "02D3}
10137 \DeclareUTFEncodedAccent\extrighthalfring{"0357}{ "02D2}
10138 \DeclareUTFDoubleEncodedSymbol\textdoublebrevebelow{"035C}{ "035C}
10139 \DeclareUTFDoubleEncodedAccent\textdoublebreve{"035D}{ "035D}
10140 \DeclareUTFDoubleEncodedAccent\textdoublemacron{"035E}{ "035E}

```

```

10141 \DeclareUTFDoubleEncodedSymbol\textdoublemacronbelow{"035F"}{"035F}
10142 \DeclareUTFDoubleEncodedAccent\textdoubletilde{"0360"}{"0360}
10143 \DeclareUTFDoubleEncodedAccent\t{"0361"}{"0361}
10144 \DeclareUTFDoubleEncodedAccent\capitaltie{"0361"}{"0361}
10145 \DeclareUTFDoubleEncodedAccent\texttoptiebar{"0361"}{"0361}
10146 \DeclareUTFDoubleEncodedSymbol\sliding{"0362"}{"0362}
10147 \DeclareUTFTIPACCommand\t
10148 \DeclareUTFTIPACCommand\capitaltie
10149 \DeclareUTFTIPACCommand\texttoptiebar
10150 \DeclareUTFTIPACCommand\sliding
10151 \DeclareUTFEncodedAccent\texthighrise{"1DC4"}{"1DC4}
10152 \DeclareUTFEncodedAccent\textlowrise{"1DC5"}{"1DC5}
10153 \DeclareUTFEncodedAccent\textrisefall{"1DC8"}{"1DC8}
10154 \DeclareUTFEncodedAccent\textfallrise{"1DC9"}{"1DC9}
10155 \DeclareUTFEncodedAccent\textaoilig{"1DD5"}{"1DD5}
10156 \DeclareUTFCompositeSymbol\textundertie{H}{"1E2A}
10157 \DeclareUTFCompositeSymbol\textundertie{h}{"1E2B}
10158 \DeclareUTFEncodedAccents\textcircumgrave{"0302"}{"0301}
10159 \DeclareUTFSymbol\textFinv{"2132}
10160 \DeclareUTFSymbol\textaleph{"2135}
10161 \DeclareUTFSymbol\textbeth{"2136}
10162 \DeclareUTFSymbol\textgimel{"2137}
10163 \DeclareUTFSymbol\textdaleth{"2138}
10164 \DeclareUTFSymbol\textGame{"2141}
10165 \DeclareUTFCompositeCommand\tonebar{25}{\tonebar{2}\tonebar{5}}
10166 \DeclareUTFCompositeCommand\tonebar{52}{\tonebar{5}\tonebar{2}}
10167 \DeclareUTFSymbol\textbigcircle{"25EF}
10168 \DeclareUTFEncodedCircle\textcircled{"20DD"}{"25EF}
10169 \DeclareUTFCompositeSymbol\textcircled{0}{"24EA}
10170 \DeclareUTFCompositeSymbol\textcircled{1}{"2460}
10171 \DeclareUTFCompositeSymbol\textcircled{2}{"2461}
10172 \DeclareUTFCompositeSymbol\textcircled{3}{"2462}
10173 \DeclareUTFCompositeSymbol\textcircled{4}{"2463}
10174 \DeclareUTFCompositeSymbol\textcircled{5}{"2464}
10175 \DeclareUTFCompositeSymbol\textcircled{6}{"2465}
10176 \DeclareUTFCompositeSymbol\textcircled{7}{"2466}
10177 \DeclareUTFCompositeSymbol\textcircled{8}{"2467}
10178 \DeclareUTFCompositeSymbol\textcircled{9}{"2468}
10179 \DeclareUTFCompositeSymbol\textcircled{10}{"2469}
10180 \DeclareUTFCompositeSymbol\textcircled{11}{"246A}
10181 \DeclareUTFCompositeSymbol\textcircled{12}{"246B}
10182 \DeclareUTFCompositeSymbol\textcircled{13}{"246C}
10183 \DeclareUTFCompositeSymbol\textcircled{14}{"246D}
10184 \DeclareUTFCompositeSymbol\textcircled{15}{"246E}
10185 \DeclareUTFCompositeSymbol\textcircled{16}{"246F}
10186 \DeclareUTFCompositeSymbol\textcircled{17}{"2470}
10187 \DeclareUTFCompositeSymbol\textcircled{18}{"2471}
10188 \DeclareUTFCompositeSymbol\textcircled{19}{"2472}
10189 \DeclareUTFCompositeSymbol\textcircled{20}{"2473}
10190 \DeclareUTFCompositeSymbol\textcircled{21}{"3251}
10191 \DeclareUTFCompositeSymbol\textcircled{22}{"3252}
10192 \DeclareUTFCompositeSymbol\textcircled{23}{"3253}
10193 \DeclareUTFCompositeSymbol\textcircled{24}{"3254}
10194 \DeclareUTFCompositeSymbol\textcircled{25}{"3255}
10195 \DeclareUTFCompositeSymbol\textcircled{26}{"3256}
10196 \DeclareUTFCompositeSymbol\textcircled{27}{"3257}
10197 \DeclareUTFCompositeSymbol\textcircled{28}{"3258}
10198 \DeclareUTFCompositeSymbol\textcircled{29}{"3259}
10199 \DeclareUTFCompositeSymbol\textcircled{30}{"325A}
10200 \DeclareUTFCompositeSymbol\textcircled{31}{"325B}
10201 \DeclareUTFCompositeSymbol\textcircled{32}{"325C}
10202 \DeclareUTFCompositeSymbol\textcircled{33}{"325D}
10203 \DeclareUTFCompositeSymbol\textcircled{34}{"325E}
10204 \DeclareUTFCompositeSymbol\textcircled{35}{"325F}
10205 \DeclareUTFCompositeSymbol\textcircled{36}{"32B1}
10206 \DeclareUTFCompositeSymbol\textcircled{37}{"32B2}
10207 \DeclareUTFCompositeSymbol\textcircled{38}{"32B3}

```

```

10208 \DeclareUTFCompositeSymbol\textcircled{39}{\textcircled{39}}{"32B4}
10209 \DeclareUTFCompositeSymbol\textcircled{40}{\textcircled{40}}{"32B5}
10210 \DeclareUTFCompositeSymbol\textcircled{41}{\textcircled{41}}{"32B6}
10211 \DeclareUTFCompositeSymbol\textcircled{42}{\textcircled{42}}{"32B7}
10212 \DeclareUTFCompositeSymbol\textcircled{43}{\textcircled{43}}{"32B8}
10213 \DeclareUTFCompositeSymbol\textcircled{44}{\textcircled{44}}{"32B9}
10214 \DeclareUTFCompositeSymbol\textcircled{45}{\textcircled{45}}{"32BA}
10215 \DeclareUTFCompositeSymbol\textcircled{46}{\textcircled{46}}{"32BB}
10216 \DeclareUTFCompositeSymbol\textcircled{47}{\textcircled{47}}{"32BC}
10217 \DeclareUTFCompositeSymbol\textcircled{48}{\textcircled{48}}{"32BD}
10218 \DeclareUTFCompositeSymbol\textcircled{49}{\textcircled{49}}{"32BE}
10219 \DeclareUTFCompositeSymbol\textcircled{50}{\textcircled{50}}{"32BF}
10220 \DeclareUTFCompositeSymbol\textcircled{A}{\textcircled{A}}{"24B6}
10221 \DeclareUTFCompositeSymbol\textcircled{B}{\textcircled{B}}{"24B7}
10222 \DeclareUTFCompositeSymbol\textcircled{C}{\textcircled{C}}{"24B8}
10223 \DeclareUTFCompositeSymbol\textcircled{D}{\textcircled{D}}{"24B9}
10224 \DeclareUTFCompositeSymbol\textcircled{E}{\textcircled{E}}{"24BA}
10225 \DeclareUTFCompositeSymbol\textcircled{F}{\textcircled{F}}{"24BB}
10226 \DeclareUTFCompositeSymbol\textcircled{G}{\textcircled{G}}{"24BC}
10227 \DeclareUTFCompositeSymbol\textcircled{H}{\textcircled{H}}{"24BD}
10228 \DeclareUTFCompositeSymbol\textcircled{I}{\textcircled{I}}{"24BE}
10229 \DeclareUTFCompositeSymbol\textcircled{J}{\textcircled{J}}{"24BF}
10230 \DeclareUTFCompositeSymbol\textcircled{K}{\textcircled{K}}{"24C0}
10231 \DeclareUTFCompositeSymbol\textcircled{L}{\textcircled{L}}{"24C1}
10232 \DeclareUTFCompositeSymbol\textcircled{M}{\textcircled{M}}{"24C2}
10233 \DeclareUTFCompositeSymbol\textcircled{N}{\textcircled{N}}{"24C3}
10234 \DeclareUTFCompositeSymbol\textcircled{O}{\textcircled{O}}{"24C4}
10235 \DeclareUTFCompositeSymbol\textcircled{P}{\textcircled{P}}{"24C5}
10236 \DeclareUTFCompositeSymbol\textcircled{Q}{\textcircled{Q}}{"24C6}
10237 \DeclareUTFCompositeSymbol\textcircled{R}{\textcircled{R}}{"24C7}
10238 \DeclareUTFCompositeSymbol\textcircled{S}{\textcircled{S}}{"24C8}
10239 \DeclareUTFCompositeSymbol\textcircled{T}{\textcircled{T}}{"24C9}
10240 \DeclareUTFCompositeSymbol\textcircled{U}{\textcircled{U}}{"24CA}
10241 \DeclareUTFCompositeSymbol\textcircled{V}{\textcircled{V}}{"24CB}
10242 \DeclareUTFCompositeSymbol\textcircled{W}{\textcircled{W}}{"24CC}
10243 \DeclareUTFCompositeSymbol\textcircled{X}{\textcircled{X}}{"24CD}
10244 \DeclareUTFCompositeSymbol\textcircled{Y}{\textcircled{Y}}{"24CE}
10245 \DeclareUTFCompositeSymbol\textcircled{Z}{\textcircled{Z}}{"24CF}
10246 \DeclareUTFCompositeSymbol\textcircled{a}{\textcircled{a}}{"24D0}
10247 \DeclareUTFCompositeSymbol\textcircled{b}{\textcircled{b}}{"24D1}
10248 \DeclareUTFCompositeSymbol\textcircled{c}{\textcircled{c}}{"24D2}
10249 \DeclareUTFCompositeSymbol\textcircled{d}{\textcircled{d}}{"24D3}
10250 \DeclareUTFCompositeSymbol\textcircled{e}{\textcircled{e}}{"24D4}
10251 \DeclareUTFCompositeSymbol\textcircled{f}{\textcircled{f}}{"24D5}
10252 \DeclareUTFCompositeSymbol\textcircled{g}{\textcircled{g}}{"24D6}
10253 \DeclareUTFCompositeSymbol\textcircled{h}{\textcircled{h}}{"24D7}
10254 \DeclareUTFCompositeSymbol\textcircled{i}{\textcircled{i}}{"24D8}
10255 \DeclareUTFCompositeSymbol\textcircled{j}{\textcircled{j}}{"24D9}
10256 \DeclareUTFCompositeSymbol\textcircled{k}{\textcircled{k}}{"24DA}
10257 \DeclareUTFCompositeSymbol\textcircled{l}{\textcircled{l}}{"24DB}
10258 \DeclareUTFCompositeSymbol\textcircled{m}{\textcircled{m}}{"24DC}
10259 \DeclareUTFCompositeSymbol\textcircled{n}{\textcircled{n}}{"24DD}
10260 \DeclareUTFCompositeSymbol\textcircled{o}{\textcircled{o}}{"24DE}
10261 \DeclareUTFCompositeSymbol\textcircled{p}{\textcircled{p}}{"24DF}
10262 \DeclareUTFCompositeSymbol\textcircled{q}{\textcircled{q}}{"24E0}
10263 \DeclareUTFCompositeSymbol\textcircled{r}{\textcircled{r}}{"24E1}
10264 \DeclareUTFCompositeSymbol\textcircled{s}{\textcircled{s}}{"24E2}
10265 \DeclareUTFCompositeSymbol\textcircled{t}{\textcircled{t}}{"24E3}
10266 \DeclareUTFCompositeSymbol\textcircled{u}{\textcircled{u}}{"24E4}
10267 \DeclareUTFCompositeSymbol\textcircled{v}{\textcircled{v}}{"24E5}
10268 \DeclareUTFCompositeSymbol\textcircled{w}{\textcircled{w}}{"24E6}
10269 \DeclareUTFCompositeSymbol\textcircled{x}{\textcircled{x}}{"24E7}
10270 \DeclareUTFCompositeSymbol\textcircled{y}{\textcircled{y}}{"24E8}
10271 \DeclareUTFCompositeSymbol\textcircled{z}{\textcircled{z}}{"24E9}
10272 \DeclareUTFCompositeSymbol\textsuperscript{h}{\textsuperscript{h}}{"02B0}
10273 \DeclareUTFCompositeSymbol\textsuperscript{\texth}{\textsuperscript{\texth}}{"02B1}
10274 \DeclareUTFCompositeSymbol\textsuperscript{j}{\textsuperscript{j}}{"02B2}

```

```

10275 \DeclareUTFCompositeSymbol\textsuperscript{r}{\02B3}
10276 \DeclareUTFCompositeSymbol\textsuperscript{\textturnr}{\02B4}
10277 \DeclareUTFCompositeSymbol\textsuperscript{\textturnrrtail}{\02B5}
10278 \DeclareUTFCompositeSymbol\textsuperscript{\textinvscr}{\02B6}
10279 \DeclareUTFCompositeSymbol\textsuperscript{w}{\02B7}
10280 \DeclareUTFCompositeSymbol\textsuperscript{y}{\02B8}
10281 \DeclareUTFCompositeSymbol\textsuperscript{\textbabygamma}{\02E0}
10282 \DeclareUTFCompositeSymbol\textsuperscript{\textgammalatinssmall}{\02E0}
10283 \DeclareUTFCompositeSymbol\textsuperscript{l}{\02E1}
10284 \DeclareUTFCompositeSymbol\textsuperscript{s}{\02E2}
10285 \DeclareUTFCompositeSymbol\textsuperscript{x}{\02E3}
10286 \DeclareUTFCompositeSymbol\textsuperscript{\textrevglotstop}{\02E4}
10287 \DeclareUTFCompositeSymbol\textsuperscript{\textrelepsilon}{\01D4C}
10288 \DeclareUTFCompositeSymbol\textsuperscript{\cyrn}{\01D78}
10289 \DeclareUTFCompositeSymbol\textsuperscript{\textbarsci}{\01DA7}
10290 \DeclareUTFCompositeSymbol\textsuperscript{V}{\02C7D}
10291 \DeclareUTFCompositeSymbol\textsuperscript{\textHbar}{\0A7F8}
10292 \DeclareUTFCompositeSymbol\textsuperscript{\textHslash}{\0A7F8}
10293 \DeclareUTFCompositeSymbol\textsuperscript{\oe}{\0A7F9}
10294 \DeclareUTFCompositeSymbol\textsubscript{h}{\02095}
10295 \DeclareUTFCompositeSymbol\textsubscript{k}{\02096}
10296 \DeclareUTFCompositeSymbol\textsubscript{l}{\02097}
10297 \DeclareUTFCompositeSymbol\textsubscript{m}{\02098}
10298 \DeclareUTFCompositeSymbol\textsubscript{n}{\02099}
10299 \DeclareUTFCompositeSymbol\textsubscript{p}{\0209A}
10300 \DeclareUTFCompositeSymbol\textsubscript{s}{\0209B}
10301 \DeclareUTFCompositeSymbol\textsubscript{t}{\0209C}

```

以下定义取自 `hyperref` 的 `puenc.def`。

```

10302 \DeclareUTFEncodedAccent\textinvbreve{\0311}{\0311}
10303 \DeclareUTFEncodedSymbol\textsubbreve{\032E}{\0203F}
10304 \DeclareUTFSymbol\textHT{\0009}
10305 \DeclareUTFSymbol\textLF{\000A}
10306 \DeclareUTFSymbol\textCR{\000D}
10307 \DeclareUTFSymbol\textnumbersign{\0023}
10308 \DeclareUTFSymbol\textparenleft{\0028}
10309 \DeclareUTFSymbol\textparenright{\0029}
10310 \DeclareUTFSymbol\textMVPlus{\002B}
10311 \DeclareUTFSymbol\textMVComma{\002C}
10312 \DeclareUTFSymbol\textMVMinus{\002D}
10313 \DeclareUTFSymbol\textMVPeriod{\002E}
10314 \DeclareUTFSymbol\textMVDivision{\002F}
10315 \DeclareUTFSymbol\textMVZero{\0030}
10316 \DeclareUTFSymbol\textMVOne{\0031}
10317 \DeclareUTFSymbol\textMVTwo{\0032}
10318 \DeclareUTFSymbol\textMVThree{\0033}
10319 \DeclareUTFSymbol\textMVFour{\0034}
10320 \DeclareUTFSymbol\textMVFive{\0035}
10321 \DeclareUTFSymbol\textMVSix{\0036}
10322 \DeclareUTFSymbol\textMVSeven{\0037}
10323 \DeclareUTFSymbol\textMVEight{\0038}
10324 \DeclareUTFSymbol\textMVNine{\0039}
10325 \DeclareUTFSymbol\textMVAt{\0040}
10326 \DeclareUTFCompositeCommand\.\{i\}{i}
10327 \DeclareUTFCompositeCommand\.\{i\}{i}
10328 \DeclareUTFSymbol\textlnot{\00AC}
10329 \DeclareUTFSymbol\textplusminus{\00B1}
10330 \DeclareUTFSymbol\textcedilla{\00B8}
10331 \DeclareUTFSymbol\textmultiply{\00D7}
10332 \DeclareUTFSymbol\textThorn{\00DE}
10333 \DeclareUTFSymbol\textdivide{\00F7}
10334 \DeclareUTFSymbol\textHslash{\0126}
10335 \DeclareUTFCompositeSymbol\k{\i}{\012F}
10336 \DeclareUTFCompositeSymbol\.\{L\}{\013F}
10337 \DeclareUTFCompositeSymbol\.\{l\}{\0140}
10338 \DeclareUTFSymbol\textnapostrophe{\0149}
10339 \DeclareUTFSymbol\textTslash{\0166}

```

```

10340 \DeclareUTFSymbol\texttslash{"0167}
10341 \DeclareUTFSymbol\textlongs{"017F}
10342 \DeclareUTFSymbol\texthausab{"0181}
10343 \DeclareUTFSymbol\texthausad{"018A}
10344 \DeclareUTFSymbol\textrevE{"018E}
10345 \DeclareUTFSymbol\texthausak{"0198}
10346 \DeclareUTFSymbol\textPUnrleg{"019E}
10347 \DeclareUTFSymbol\textinve{"01DD}
10348 \DeclareUTFSymbol\textGslash{"01E4}
10349 \DeclareUTFSymbol\textgslash{"01E5}
10350 \DeclareUTFCompositeSymbol\textinvbreve{E}{{"0206}
10351 \DeclareUTFCompositeSymbol\textinvbreve{e}{{"0207}
10352 \DeclareUTFCompositeSymbol\textinvbreve{I}{{"020A}
10353 \DeclareUTFCompositeSymbol\textinvbreve{i}{{"020B}
10354 \DeclareUTFCompositeSymbol\textinvbreve{I}{{"020B}
10355 \DeclareUTFCompositeSymbol\textinvbreve{O}{{"020E}
10356 \DeclareUTFCompositeSymbol\textinvbreve{o}{{"020F}
10357 \DeclareUTFCompositeSymbol\textinvbreve{U}{{"0216}
10358 \DeclareUTFCompositeSymbol\textinvbreve{u}{{"0217}
10359 \DeclareUTFSymbol\j{"0237}
10360 \DeclareUTFSymbol\textPUdblig{"0238}
10361 \DeclareUTFSymbol\textPUqplig{"0239}
10362 \DeclareUTFSymbol\textslashc{"023C}
10363 \DeclareUTFSymbol\textniepsilon{"025B}
10364 \DeclareUTFSymbol\textipagamma{"0263}
10365 \DeclareUTFSymbol\textniiota{"0269}
10366 \DeclareUTFSymbol\textnippi{"0278}
10367 \DeclareUTFSymbol\textniupsilon{"028A}
10368 \DeclareUTFSymbol\textring{"02DA}
10369 \DeclareUTFSymbol\texttilde{"02DC}
10370 \DeclareUTFSymbol\texthungarumlaut{"02DD}
10371 \DeclareUTFSymbol\textringlow{"02F3}
10372 \DeclareUTFSymbol\texttildelow{"02F7}
10373 \DeclareUTFCommand\textnewtie{\textinvbreve\ }
10374 \DeclareUTFCommand\textdotbelow{\d\ }
10375 \DeclareUTFSymbol\textmacronbelow{"02CD}
10376 \DeclareUTFCommand\texttie{\t\ }
10377 \DeclareUTFSymbol\textnumeralsigngreek{"0374}
10378 \DeclareUTFSymbol\textnumeralsignlowergreek{"0375}
10379 \DeclareUTFCompositeSymbol\'\textAlpha{"0386}
10380 \DeclareUTFCompositeSymbol\'\textEpsilon{"0388}
10381 \DeclareUTFCompositeSymbol\'\textEta{"0389}
10382 \DeclareUTFCompositeSymbol\'\textIota{"038A}
10383 \DeclareUTFCompositeSymbol\'\textOmicron{"038C}
10384 \DeclareUTFCompositeSymbol\'\textUpsilon{"038E}
10385 \DeclareUTFCompositeSymbol\'\textOmega{"038F}
10386 \DeclareUTFCompositeSymbol\'\textIotadieresis{"0390}
10387 \DeclareUTFSymbol\textIotadieresis{"03AA}
10388 \DeclareUTFCompositeSymbol\'\textIota{"03AA}
10389 \DeclareUTFCompositeSymbol\'\textUpsilon{"03AB}
10390 \DeclareUTFCompositeSymbol\'\textalpha{"03AC}
10391 \DeclareUTFCompositeSymbol\'\textepsilon{"03AD}
10392 \DeclareUTFCompositeSymbol\'\texteta{"03AE}
10393 \DeclareUTFCompositeSymbol\'\textiota{"03AF}
10394 \DeclareUTFCompositeSymbol\'\textupsilonacute{"03B0}
10395 \DeclareUTFSymbol\textmugreek{"03BC}
10396 \DeclareUTFSymbol\textvarsigma{"03C2}
10397 \DeclareUTFCompositeSymbol\'\textiota{"03CA}
10398 \DeclareUTFCompositeSymbol\'\textupsilon{"03CB}
10399 \DeclareUTFCompositeSymbol\'\textomicron{"03CC}
10400 \DeclareUTFSymbol\textupsilonacute{"03CD}
10401 \DeclareUTFCompositeSymbol\'\textupsilon{"03CD}
10402 \DeclareUTFCompositeSymbol\'\textomega{"03CE}
10403 \DeclareUTFSymbol\textStigmagreek{"03DA}
10404 \DeclareUTFSymbol\textstigmagreek{"03DB}
10405 \DeclareUTFSymbol\textDigammagreek{"03DC}
10406 \DeclareUTFSymbol\textdigammagreek{"03DD}

```

```

10407 \DeclareUTFSymbol\textKoppagreek{"03DE}
10408 \DeclareUTFSymbol\textkoppagreek{"03DF}
10409 \DeclareUTFSymbol\textSampigreek{"03E0}
10410 \DeclareUTFSymbol\textsampigreek{"03E1}
10411 \DeclareUTFSymbol\textbackepsilon{"03F6}
10412 \DeclareUTFCompositeSymbol\`{\CYRE}{0400}
10413 \DeclareUTFSymbol\CYRYO{"0401}
10414 \DeclareUTFCompositeSymbol\`{\CYRE}{0401}
10415 \DeclareUTFSymbol\CYRDJE{"0402}
10416 \DeclareUTFCompositeSymbol\`{\CYRG}{0403}
10417 \DeclareUTFSymbol\CYRIE{"0404}
10418 \DeclareUTFSymbol\CYRDZE{"0405}
10419 \DeclareUTFSymbol\CYRII{"0406}
10420 \DeclareUTFSymbol\CYRYI{"0407}
10421 \DeclareUTFCompositeSymbol\`{\CYRII}{0407}
10422 \DeclareUTFSymbol\CYRJE{"0408}
10423 \DeclareUTFSymbol\CYRLJE{"0409}
10424 \DeclareUTFSymbol\CYRNJE{"040A}
10425 \DeclareUTFSymbol\CYRTSHE{"040B}
10426 \DeclareUTFCompositeSymbol\`{\CYRK}{040C}
10427 \DeclareUTFCompositeSymbol\`{\CYRI}{040D}
10428 \DeclareUTFSymbol\CYRUSHRT{"040E}
10429 \DeclareUTFCompositeSymbol\U{\CYRU}{040E}
10430 \DeclareUTFSymbol\CYRDZHE{"040F}
10431 \DeclareUTFSymbol\CYRA{"0410}
10432 \DeclareUTFSymbol\CYRB{"0411}
10433 \DeclareUTFSymbol\CYRV{"0412}
10434 \DeclareUTFSymbol\CYRG{"0413}
10435 \DeclareUTFSymbol\CYRD{"0414}
10436 \DeclareUTFSymbol\CYRE{"0415}
10437 \DeclareUTFSymbol\CYRZH{"0416}
10438 \DeclareUTFSymbol\CYRZ{"0417}
10439 \DeclareUTFSymbol\CYRI{"0418}
10440 \DeclareUTFSymbol\CYRISHRT{"0419}
10441 \DeclareUTFCompositeSymbol\U{\CYRI}{0419}
10442 \DeclareUTFSymbol\CYRK{"041A}
10443 \DeclareUTFSymbol\CYRL{"041B}
10444 \DeclareUTFSymbol\CYRM{"041C}
10445 \DeclareUTFSymbol\CYRN{"041D}
10446 \DeclareUTFSymbol\CYRO{"041E}
10447 \DeclareUTFSymbol\CYRP{"041F}
10448 \DeclareUTFSymbol\CYRR{"0420}
10449 \DeclareUTFSymbol\CYRS{"0421}
10450 \DeclareUTFSymbol\CYRT{"0422}
10451 \DeclareUTFSymbol\CYRU{"0423}
10452 \DeclareUTFSymbol\CYRF{"0424}
10453 \DeclareUTFSymbol\CYRH{"0425}
10454 \DeclareUTFSymbol\CYRC{"0426}
10455 \DeclareUTFSymbol\CYRCH{"0427}
10456 \DeclareUTFSymbol\CYRSH{"0428}
10457 \DeclareUTFSymbol\CYRSHCH{"0429}
10458 \DeclareUTFSymbol\CYRHRDSN{"042A}
10459 \DeclareUTFSymbol\CYRERY{"042B}
10460 \DeclareUTFSymbol\CYRSFTSN{"042C}
10461 \DeclareUTFSymbol\CYREREV{"042D}
10462 \DeclareUTFSymbol\CYRYU{"042E}
10463 \DeclareUTFSymbol\CYRYA{"042F}
10464 \DeclareUTFSymbol\cyra{"0430}
10465 \DeclareUTFSymbol\cyrb{"0431}
10466 \DeclareUTFSymbol\cyrv{"0432}
10467 \DeclareUTFSymbol\cyrg{"0433}
10468 \DeclareUTFSymbol\cyrd{"0434}
10469 \DeclareUTFSymbol\cyre{"0435}
10470 \DeclareUTFSymbol\cyrz{"0436}
10471 \DeclareUTFSymbol\cyrz{"0437}
10472 \DeclareUTFSymbol\cyri{"0438}
10473 \DeclareUTFSymbol\cyrishrt{"0439}

```

```

10474 \DeclareUTFCompositeSymbol\U{\cyri}{"0439}
10475 \DeclareUTFSymbol\cyrk{"043A}
10476 \DeclareUTFSymbol\cyr1{"043B}
10477 \DeclareUTFSymbol\cyrn{"043C}
10478 \DeclareUTFSymbol\cyrn{"043D}
10479 \DeclareUTFSymbol\cyro{"043E}
10480 \DeclareUTFSymbol\cyrp{"043F}
10481 \DeclareUTFSymbol\cyrr{"0440}
10482 \DeclareUTFSymbol\cyrs{"0441}
10483 \DeclareUTFSymbol\cyrt{"0442}
10484 \DeclareUTFSymbol\cyr1{"0443}
10485 \DeclareUTFSymbol\cyrf{"0444}
10486 \DeclareUTFSymbol\cyrh{"0445}
10487 \DeclareUTFSymbol\cyr1{"0446}
10488 \DeclareUTFSymbol\cyrch{"0447}
10489 \DeclareUTFSymbol\cyrsh{"0448}
10490 \DeclareUTFSymbol\cyrshch{"0449}
10491 \DeclareUTFSymbol\cyrhrdsn{"044A}
10492 \DeclareUTFSymbol\cyrry{"044B}
10493 \DeclareUTFSymbol\cyrsftsn{"044C}
10494 \DeclareUTFSymbol\cyrrrev{"044D}
10495 \DeclareUTFSymbol\cyr1{"044E}
10496 \DeclareUTFSymbol\cyr1{"044F}
10497 \DeclareUTFCompositeSymbol\`{\cyre}{"0450}
10498 \DeclareUTFSymbol\cyr1{"0451}
10499 \DeclareUTFCompositeSymbol\`{\cyre}{"0451}
10500 \DeclareUTFSymbol\cyrdje{"0452}
10501 \DeclareUTFCompositeSymbol\`{\cyrg}{"0453}
10502 \DeclareUTFSymbol\cyrie{"0454}
10503 \DeclareUTFSymbol\cyrdze{"0455}
10504 \DeclareUTFSymbol\cyrii{"0456}
10505 \DeclareUTFSymbol\cyr1{"0457}
10506 \DeclareUTFCompositeSymbol\`{\cyrii}{"0457}
10507 \DeclareUTFSymbol\cyrje{"0458}
10508 \DeclareUTFSymbol\cyr1je{"0459}
10509 \DeclareUTFSymbol\cyrnje{"045A}
10510 \DeclareUTFSymbol\cyrtsh{"045B}
10511 \DeclareUTFCompositeSymbol\`{\cyrk}{"045C}
10512 \DeclareUTFCompositeSymbol\`{\cyri}{"045D}
10513 \DeclareUTFSymbol\cyrushrt{"045E}
10514 \DeclareUTFCompositeSymbol\U{\curu}{"045E}
10515 \DeclareUTFSymbol\cyrdzhe{"045F}
10516 \DeclareUTFSymbol\CYROMEGA{"0460}
10517 \DeclareUTFSymbol\cyromega{"0461}
10518 \DeclareUTFSymbol\CYRYAT{"0462}
10519 \DeclareUTFSymbol\cyr1{"0463}
10520 \DeclareUTFSymbol\CYRIOTE{"0464}
10521 \DeclareUTFSymbol\cyriote{"0465}
10522 \DeclareUTFSymbol\CYRLYUS{"0466}
10523 \DeclareUTFSymbol\cyrlyus{"0467}
10524 \DeclareUTFSymbol\CYRIOTLYUS{"0468}
10525 \DeclareUTFSymbol\cyriotlyus{"0469}
10526 \DeclareUTFSymbol\CYRBYUS{"046A}
10527 \DeclareUTFSymbol\cyrbyus{"046B}
10528 \DeclareUTFSymbol\CYRIOTBYUS{"046C}
10529 \DeclareUTFSymbol\cyriotbyus{"046D}
10530 \DeclareUTFSymbol\CYRKSI{"046E}
10531 \DeclareUTFSymbol\cyrksi{"046F}
10532 \DeclareUTFSymbol\CYRPSI{"0470}
10533 \DeclareUTFSymbol\cyrpsi{"0471}
10534 \DeclareUTFSymbol\CYRFITA{"0472}
10535 \DeclareUTFSymbol\cyrfita{"0473}
10536 \DeclareUTFSymbol\CYRIZH{"0474}
10537 \DeclareUTFSymbol\cyrizh{"0475}
10538 \DeclareUTFCompositeSymbol\C{\CYRIZH}{"0476}
10539 \DeclareUTFCompositeSymbol\C{\cyrizh}{"0477}
10540 \DeclareUTFSymbol\CYRUK{"0478}

```

```

10541 \DeclareUTFSymbol\cyruk{"0479}
10542 \DeclareUTFSymbol\CYROMEGARND{"047A}
10543 \DeclareUTFSymbol\cyromegarnd{"047B}
10544 \DeclareUTFSymbol\CYROMEGATITLO{"047C}
10545 \DeclareUTFSymbol\cyromegatitlo{"047D}
10546 \DeclareUTFSymbol\CYROT{"047E}
10547 \DeclareUTFSymbol\cyrot{"047F}
10548 \DeclareUTFSymbol\CYRKOPPA{"0480}
10549 \DeclareUTFSymbol\cyrkoppa{"0481}
10550 \DeclareUTFSymbol\cyrthousands{"0482}
10551 \DeclareUTFSymbol\CYRISHRTDSC{"048A}
10552 \DeclareUTFSymbol\cyrishrtdsc{"048B}
10553 \DeclareUTFSymbol\CYRSEMISFTSN{"048C}
10554 \DeclareUTFSymbol\cyrsemisftsn{"048D}
10555 \DeclareUTFSymbol\CYRRTICK{"048E}
10556 \DeclareUTFSymbol\cyrrtick{"048F}
10557 \DeclareUTFSymbol\CYRGUP{"0490}
10558 \DeclareUTFSymbol\cyrgup{"0491}
10559 \DeclareUTFSymbol\CYRGHCRS{"0492}
10560 \DeclareUTFSymbol\cyrghcrs{"0493}
10561 \DeclareUTFSymbol\CYRGHK{"0494}
10562 \DeclareUTFSymbol\cyrghk{"0495}
10563 \DeclareUTFSymbol\CYRZHDSC{"0496}
10564 \DeclareUTFSymbol\cyrzhdsc{"0497}
10565 \DeclareUTFSymbol\CYRZDSC{"0498}
10566 \DeclareUTFCompositeSymbol\c{\CYRZ}{0498}
10567 \DeclareUTFSymbol\cyrzdsc{"0499}
10568 \DeclareUTFCompositeSymbol\c{\cyrz}{0499}
10569 \DeclareUTFSymbol\CYRKDSC{"049A}
10570 \DeclareUTFSymbol\cyrkdsc{"049B}
10571 \DeclareUTFSymbol\CYRKVCRS{"049C}
10572 \DeclareUTFSymbol\cyrkvcrs{"049D}
10573 \DeclareUTFSymbol\CYRKHCRS{"049E}
10574 \DeclareUTFSymbol\cyrkhcrs{"049F}
10575 \DeclareUTFSymbol\CYRKBEAK{"04A0}
10576 \DeclareUTFSymbol\cyrkbeak{"04A1}
10577 \DeclareUTFSymbol\CYRNDSC{"04A2}
10578 \DeclareUTFSymbol\cyrndsc{"04A3}
10579 \DeclareUTFSymbol\CYRNG{"04A4}
10580 \DeclareUTFSymbol\cyrng{"04A5}
10581 \DeclareUTFSymbol\CYRPHK{"04A6}
10582 \DeclareUTFSymbol\cyrphk{"04A7}
10583 \DeclareUTFSymbol\CYRABHHA{"04A8}
10584 \DeclareUTFSymbol\cyrabhha{"04A9}
10585 \DeclareUTFSymbol\CYRSDSC{"04AA}
10586 \DeclareUTFCompositeSymbol\k{\CYRS}{04AA}
10587 \DeclareUTFSymbol\cyrsdsc{"04AB}
10588 \DeclareUTFCompositeSymbol\k{\cyrs}{04AB}
10589 \DeclareUTFSymbol\CYRTDSC{"04AC}
10590 \DeclareUTFSymbol\cyrttdsc{"04AD}
10591 \DeclareUTFSymbol\CYRY{"04AE}
10592 \DeclareUTFSymbol\cyry{"04AF}
10593 \DeclareUTFSymbol\CYRYHCRS{"04B0}
10594 \DeclareUTFSymbol\cyryhcrs{"04B1}
10595 \DeclareUTFSymbol\CYRHDSC{"04B2}
10596 \DeclareUTFSymbol\cyrhdsc{"04B3}
10597 \DeclareUTFSymbol\CYRTETSE{"04B4}
10598 \DeclareUTFSymbol\cyrtetse{"04B5}
10599 \DeclareUTFSymbol\CYRCHRDSC{"04B6}
10600 \DeclareUTFSymbol\cyrchrdsc{"04B7}
10601 \DeclareUTFSymbol\CYRCHVCRS{"04B8}
10602 \DeclareUTFSymbol\cyrchvcrs{"04B9}
10603 \DeclareUTFSymbol\CYRSHHA{"04BA}
10604 \DeclareUTFSymbol\cyrshha{"04BB}
10605 \DeclareUTFSymbol\CYRABHCH{"04BC}
10606 \DeclareUTFSymbol\cyrabhch{"04BD}
10607 \DeclareUTFSymbol\CYRABHCHDSC{"04BE}

```

```

10608 \DeclareUTFCompositeSymbol\k{\CYRABHCH}{04BE}
10609 \DeclareUTFSymbol\cyrahchdsc{04BF}
10610 \DeclareUTFCompositeSymbol\k{\cyrahch}{04BF}
10611 \DeclareUTFSymbol\CYRpalochnka{04C0}
10612 \DeclareUTFCompositeSymbol\U{\CYRZH}{04C1}
10613 \DeclareUTFCompositeSymbol\U{\cyrz}{04C2}
10614 \DeclareUTFSymbol\CYRKHK{04C3}
10615 \DeclareUTFSymbol\cyrkhhk{04C4}
10616 \DeclareUTFSymbol\CYRLDSC{04C5}
10617 \DeclareUTFSymbol\cyrlldsc{04C6}
10618 \DeclareUTFSymbol\CYRNHK{04C7}
10619 \DeclareUTFSymbol\cyrnhhk{04C8}
10620 \DeclareUTFSymbol\CYRCHLDSC{04CB}
10621 \DeclareUTFSymbol\cyrchldsc{04CC}
10622 \DeclareUTFSymbol\CYRMDSC{04CD}
10623 \DeclareUTFSymbol\cyrmdsc{04CE}
10624 \DeclareUTFCompositeSymbol\U{\CYRA}{04D0}
10625 \DeclareUTFCompositeSymbol\U{\cyra}{04D1}
10626 \DeclareUTFCompositeSymbol\{\CYRA}{04D2}
10627 \DeclareUTFCompositeSymbol\{\cyra}{04D3}
10628 \DeclareUTFSymbol\CYRAE{04D4}
10629 \DeclareUTFSymbol\cyrae{04D5}
10630 \DeclareUTFCompositeSymbol\U{\CYRE}{04D6}
10631 \DeclareUTFCompositeSymbol\U{\cyre}{04D7}
10632 \DeclareUTFSymbol\CYRSCHWA{04D8}
10633 \DeclareUTFSymbol\cyrschwa{04D9}
10634 \DeclareUTFCompositeSymbol\{\CYRSCHWA}{04DA}
10635 \DeclareUTFCompositeSymbol\{\cyrschwa}{04DB}
10636 \DeclareUTFCompositeSymbol\{\CYRZH}{04DC}
10637 \DeclareUTFCompositeSymbol\{\cyrz}{04DD}
10638 \DeclareUTFCompositeSymbol\{\CYRZ}{04DE}
10639 \DeclareUTFCompositeSymbol\{\cyrz}{04DF}
10640 \DeclareUTFSymbol\CYRABHDZE{04E0}
10641 \DeclareUTFSymbol\cyrahhdze{04E1}
10642 \DeclareUTFCompositeSymbol\{\CYRI}{04E2}
10643 \DeclareUTFCompositeSymbol\{\cyri}{04E3}
10644 \DeclareUTFCompositeSymbol\{\CYRI}{04E4}
10645 \DeclareUTFCompositeSymbol\{\cyri}{04E5}
10646 \DeclareUTFCompositeSymbol\{\CYRO}{04E6}
10647 \DeclareUTFCompositeSymbol\{\cyro}{04E7}
10648 \DeclareUTFSymbol\CYROTLD{04E8}
10649 \DeclareUTFSymbol\cyrotld{04E9}
10650 \DeclareUTFCompositeSymbol\{\CYROTLD}{04EA}
10651 \DeclareUTFCompositeSymbol\{\cyrotld}{04EB}
10652 \DeclareUTFCompositeSymbol\{\CYREREV}{04EC}
10653 \DeclareUTFCompositeSymbol\{\cyrerev}{04ED}
10654 \DeclareUTFCompositeSymbol\{\CYRU}{04EE}
10655 \DeclareUTFCompositeSymbol\{\cyru}{04EF}
10656 \DeclareUTFCompositeSymbol\{\CYRU}{04F0}
10657 \DeclareUTFCompositeSymbol\{\cyru}{04F1}
10658 \DeclareUTFCompositeSymbol\H{\CYRU}{04F2}
10659 \DeclareUTFCompositeSymbol\H{\cyru}{04F3}
10660 \DeclareUTFCompositeSymbol\{\CYRCH}{04F4}
10661 \DeclareUTFCompositeSymbol\{\cyrch}{04F5}
10662 \DeclareUTFSymbol\CYRGDSC{04F6}
10663 \DeclareUTFSymbol\cyrgdsc{04F7}
10664 \DeclareUTFCompositeSymbol\{\CYRERY}{04F8}
10665 \DeclareUTFCompositeSymbol\{\cyrery}{04F9}
10666 \DeclareUTFSymbol\CYRHHK{04FC}
10667 \DeclareUTFSymbol\cyrhkhk{04FD}
10668 \DeclareUTFSymbol\sofpasuq{05C3}
10669 \DeclareUTFSymbol\hebalef{05D0}
10670 \DeclareUTFSymbol\hebbet{05D1}
10671 \DeclareUTFSymbol\hebgimel{05D2}
10672 \DeclareUTFSymbol\hebdalet{05D3}
10673 \DeclareUTFSymbol\hebhe{05D4}
10674 \DeclareUTFSymbol\hebvav{05D5}

```

```

10675 \DeclareUTFSymbol\hebzayin{"05D6}
10676 \DeclareUTFSymbol\hebhet{"05D7}
10677 \DeclareUTFSymbol\hebtet{"05D8}
10678 \DeclareUTFSymbol\hebyod{"05D9}
10679 \DeclareUTFSymbol\hebfinalkaf{"05DA}
10680 \DeclareUTFSymbol\hebkafe{"05DB}
10681 \DeclareUTFSymbol\heblamed{"05DC}
10682 \DeclareUTFSymbol\hebfinalmem{"05DD}
10683 \DeclareUTFSymbol\hebmeme{"05DE}
10684 \DeclareUTFSymbol\hebfinalnun{"05DF}
10685 \DeclareUTFSymbol\hebnun{"05E0}
10686 \DeclareUTFSymbol\hebsamekh{"05E1}
10687 \DeclareUTFSymbol\hebayin{"05E2}
10688 \DeclareUTFSymbol\hebfinalpe{"05E3}
10689 \DeclareUTFSymbol\hebpe{"05E4}
10690 \DeclareUTFSymbol\hebfinaltsadi{"05E5}
10691 \DeclareUTFSymbol\hebtsadi{"05E6}
10692 \DeclareUTFSymbol\hebqof{"05E7}
10693 \DeclareUTFSymbol\hebresh{"05E8}
10694 \DeclareUTFSymbol\hebshin{"05E9}
10695 \DeclareUTFSymbol\hebtav{"05EA}
10696 \DeclareUTFSymbol\doublevav{"05F0}
10697 \DeclareUTFSymbol\vavyod{"05F1}
10698 \DeclareUTFSymbol\doubleyod{"05F2}
10699 \DeclareUTFSymbol\textscd{"1D05}
10700 \DeclareUTFSymbol\textPUsck{"1D0B}
10701 \DeclareUTFSymbol\textPUscm{"1D0D}
10702 \DeclareUTFSymbol\textPUscp{"1D18}
10703 \DeclareUTFSymbol\textPUrevscr{"1D19}
10704 \DeclareUTFSymbol\textiinferior{"1D62}
10705 \DeclareUTFSymbol\textrinferior{"1D63}
10706 \DeclareUTFSymbol\textuinferior{"1D64}
10707 \DeclareUTFSymbol\textvinferior{"1D65}
10708 \DeclareUTFSymbol\textbetainferior{"1D66}
10709 \DeclareUTFSymbol\textgammainferior{"1D67}
10710 \DeclareUTFSymbol\textrhoinferior{"1D68}
10711 \DeclareUTFSymbol\textphiinferior{"1D69}
10712 \DeclareUTFSymbol\textchiinferior{"1D6A}
10713 \DeclareUTFSymbol\textbarsci{"1D7B}
10714 \DeclareUTFSymbol\textbarp{"1D7D}
10715 \DeclareUTFSymbol\textbarscu{"1D7E}
10716 \DeclareUTFSymbol\textPURhooka{"1D8F}
10717 \DeclareUTFSymbol\textPURhooke{"1D92}
10718 \DeclareUTFSymbol\textPURhookepsilon{"1D93}
10719 \DeclareUTFSymbol\textPURhookopeno{"1D97}
10720 \DeclareUTFCompositeSymbol\textsubbreve{H}{{"1E2A}
10721 \DeclareUTFCompositeSymbol\textsubbreve{h}{{"1E2B}
10722 \DeclareUTFCompositeSymbol\.\{\textlong}{{"1E9B}
10723 \DeclareUTFSymbol\textcompwordmark{"200C}
10724 \DeclareUTFSymbol\texthdotfor{"2025}
10725 \DeclareUTFSymbol\textprime{"2032}
10726 \DeclareUTFSymbol\textsecond{"2033}
10727 \DeclareUTFSymbol\textthird{"2034}
10728 \DeclareUTFSymbol\textbackprime{"2035}
10729 \DeclareUTFSymbol\textlefttherefore{"2056}
10730 \DeclareUTFSymbol\textfourth{"2057}
10731 \DeclareUTFSymbol\textdiamonddots{"2058}
10732 \DeclareUTFSymbol\textzerosuperior{"2070}
10733 \DeclareUTFSymbol\textisuperior{"2071}
10734 \DeclareUTFSymbol\textfoursuperior{"2074}
10735 \DeclareUTFSymbol\textfivesuperior{"2075}
10736 \DeclareUTFSymbol\textsixsuperior{"2076}
10737 \DeclareUTFSymbol\textsevensuperior{"2077}
10738 \DeclareUTFSymbol\texteightsuperior{"2078}
10739 \DeclareUTFSymbol\textninesuperior{"2079}
10740 \DeclareUTFSymbol\textplussuperior{"207A}
10741 \DeclareUTFSymbol\textminussuperior{"207B}

```

```

10742 \DeclareUTFSymbol\textequalsuperior{"207C}
10743 \DeclareUTFSymbol\textparenleftsuperior{"207D}
10744 \DeclareUTFSymbol\textparenrightsuperior{"207E}
10745 \DeclareUTFSymbol\textnsuperior{"207F}
10746 \DeclareUTFSymbol\textzeroinferior{"2080}
10747 \DeclareUTFSymbol\textoneinferior{"2081}
10748 \DeclareUTFSymbol\texttwoinferior{"2082}
10749 \DeclareUTFSymbol\textthreeinferior{"2083}
10750 \DeclareUTFSymbol\textfourinferior{"2084}
10751 \DeclareUTFSymbol\textfiveinferior{"2085}
10752 \DeclareUTFSymbol\textsixinferior{"2086}
10753 \DeclareUTFSymbol\textseveninferior{"2087}
10754 \DeclareUTFSymbol\texteightinferior{"2088}
10755 \DeclareUTFSymbol\textnineinferior{"2089}
10756 \DeclareUTFSymbol\textplusinferior{"208A}
10757 \DeclareUTFSymbol\textminusinferior{"208B}
10758 \DeclareUTFSymbol\textequalsinferior{"208C}
10759 \DeclareUTFSymbol\textparenleftinferior{"208D}
10760 \DeclareUTFSymbol\textparenrightinferior{"208E}
10761 \DeclareUTFSymbol\textainferior{"2090}
10762 \DeclareUTFSymbol\texteinferior{"2091}
10763 \DeclareUTFSymbol\textoinferior{"2092}
10764 \DeclareUTFSymbol\textxinferior{"2093}
10765 \DeclareUTFSymbol\textschwainferior{"2094}
10766 \DeclareUTFSymbol\texthinferior{"2095}
10767 \DeclareUTFSymbol\textkinferior{"2096}
10768 \DeclareUTFSymbol\textlinferior{"2097}
10769 \DeclareUTFSymbol\textminferior{"2098}
10770 \DeclareUTFSymbol\textninferior{"2099}
10771 \DeclareUTFSymbol\textpinferior{"209A}
10772 \DeclareUTFSymbol\textsinferior{"209B}
10773 \DeclareUTFSymbol\texttinferior{"209C}
10774 \DeclareUTFSymbol\textpeseta{"20A7}
10775 \DeclareUTFSymbol\textDeleatur{"20B0}
10776 \DeclareUTFSymbol\textguarani{"20B2}
10777 \DeclareUTFSymbol\texthslash{"210F}
10778 \DeclareUTFSymbol\textlm{"2111}
10779 \DeclareUTFSymbol\textell{"2113}
10780 \DeclareUTFSymbol\textwp{"2118}
10781 \DeclareUTFSymbol\textRe{"211C}
10782 \DeclareUTFSymbol\textriota{"2129}
10783 \DeclareUTFSymbol\textangstrom{"212B}
10784 \DeclareUTFSymbol\textfax{"213B}
10785 \DeclareUTFSymbol\textinvamp{"214B}
10786 \DeclareUTFSymbol\textoneseventh{"2150}
10787 \DeclareUTFSymbol\textoneninth{"2151}
10788 \DeclareUTFSymbol\textonetenth{"2152}
10789 \DeclareUTFSymbol\textonethird{"2153}
10790 \DeclareUTFSymbol\texttwothirds{"2154}
10791 \DeclareUTFSymbol\textonefifth{"2155}
10792 \DeclareUTFSymbol\texttwofifths{"2156}
10793 \DeclareUTFSymbol\textthreefifths{"2157}
10794 \DeclareUTFSymbol\textfourfifths{"2158}
10795 \DeclareUTFSymbol\textonesixth{"2159}
10796 \DeclareUTFSymbol\textfivesixths{"215A}
10797 \DeclareUTFSymbol\textoneeighth{"215B}
10798 \DeclareUTFSymbol\textthreeeighths{"215C}
10799 \DeclareUTFSymbol\textfiveeighths{"215D}
10800 \DeclareUTFSymbol\textseveneighths{"215E}
10801 \DeclareUTFSymbol\textrevc{"2184}
10802 \DeclareUTFSymbol\textzerothirds{"2189}
10803 \DeclareUTFSymbol\textnleftarrow{"219A}
10804 \DeclareUTFSymbol\textnrightarrow{"219B}
10805 \DeclareUTFSymbol\texttwoheadleftarrow{"219E}
10806 \DeclareUTFCommand\textntwoheadleftarrow{\textlstrikethru\texttwoheadleftarrow}
10807 \DeclareUTFSymbol\texttwoheaduparrow{"219F}
10808 \DeclareUTFSymbol\texttwoheadrightarrow{"21A0}

```

```

10809 \DeclareUTFCommand\textntwoheadrightarrow{\textlstrikethru\texttwoheadrightarrow}
10810 \DeclareUTFSymbol\texttwoheaddownarrow{"21A1}
10811 \DeclareUTFSymbol\textleftarrowtail{"21A2}
10812 \DeclareUTFSymbol\textrightarrowtail{"21A3}
10813 \DeclareUTFSymbol\textmapsto{"21A6}
10814 \DeclareUTFSymbol\texthookleftarrow{"21A9}
10815 \DeclareUTFSymbol\texthookrightarrow{"21AA}
10816 \DeclareUTFSymbol\textlooparrowleft{"21AB}
10817 \DeclareUTFSymbol\textlooparrowright{"21AC}
10818 \DeclareUTFSymbol\textnleftrightarrow{"21AE}
10819 \DeclareUTFSymbol\textlightning{"21AF}
10820 \DeclareUTFSymbol\textdsh{"21B5}
10821 \DeclareUTFSymbol\textcurvearrowleft{"21B6}
10822 \DeclareUTFSymbol\textcurvearrowright{"21B7}
10823 \DeclareUTFSymbol\textleftharpoonup{"21BC}
10824 \DeclareUTFSymbol\textleftharpoondown{"21BD}
10825 \DeclareUTFSymbol\textupharpoonright{"21BE}
10826 \DeclareUTFSymbol\textupharpoonleft{"21BF}
10827 \DeclareUTFSymbol\texttrightharpoonup{"21C0}
10828 \DeclareUTFSymbol\texttrightharpoondown{"21C1}
10829 \DeclareUTFSymbol\textdownharpoonright{"21C2}
10830 \DeclareUTFSymbol\textdownharpoonleft{"21C3}
10831 \DeclareUTFSymbol\textrightleftarrows{"21C4}
10832 \DeclareUTFSymbol\textupdownarrows{"21C5}
10833 \DeclareUTFSymbol\textleftrightharrows{"21C6}
10834 \DeclareUTFSymbol\textleftleftarrows{"21C7}
10835 \DeclareUTFSymbol\textupuparrows{"21C8}
10836 \DeclareUTFSymbol\texttrightrightarrows{"21C9}
10837 \DeclareUTFSymbol\textdowndownarrows{"21CA}
10838 \DeclareUTFSymbol\textleftrightharpoons{"21CB}
10839 \DeclareUTFSymbol\textrightleftharpoons{"21CC}
10840 \DeclareUTFSymbol\textnLeftarrow{"21CD}
10841 \DeclareUTFSymbol\textnLeftrightarrow{"21CE}
10842 \DeclareUTFSymbol\textnRightarrow{"21CF}
10843 \DeclareUTFSymbol\textLeftarrow{"21D0}
10844 \DeclareUTFSymbol\textUparrow{"21D1}
10845 \DeclareUTFSymbol\textRightarrow{"21D2}
10846 \DeclareUTFSymbol\textDownarrow{"21D3}
10847 \DeclareUTFSymbol\textLeftrightarrow{"21D4}
10848 \DeclareUTFSymbol\textUpdownarrow{"21D5}
10849 \DeclareUTFSymbol\textNarrow{"21D6}
10850 \DeclareUTFSymbol\textNearrow{"21D7}
10851 \DeclareUTFSymbol\textSearrow{"21D8}
10852 \DeclareUTFSymbol\textSwarrow{"21D9}
10853 \DeclareUTFSymbol\textLleftarrow{"21DA}
10854 \DeclareUTFSymbol\textRrightarrow{"21DB}
10855 \DeclareUTFSymbol\textleftsquigarrow{"21DC}
10856 \DeclareUTFSymbol\textright squigarrow{"21DD}
10857 \DeclareUTFSymbol\textdashleftarrow{"21E0}
10858 \DeclareUTFSymbol\textdasheduparrow{"21E1}
10859 \DeclareUTFSymbol\textdashrightarrow{"21E2}
10860 \DeclareUTFSymbol\textdasheddownarrow{"21E3}
10861 \DeclareUTFSymbol\textpointer{"21E8}
10862 \DeclareUTFSymbol\textdownuparrows{"21F5}
10863 \DeclareUTFSymbol\textleftarrowtriangle{"21FD}
10864 \DeclareUTFSymbol\textrightarrowtriangle{"21FE}
10865 \DeclareUTFSymbol\textleftrightharpoontriangle{"21FF}
10866 \DeclareUTFSymbol\textforall{"2200}
10867 \DeclareUTFSymbol\textcomplement{"2201}
10868 \DeclareUTFSymbol\textpartial{"2202}
10869 \DeclareUTFSymbol\textexists{"2203}
10870 \DeclareUTFSymbol\textnexists{"2204}
10871 \DeclareUTFSymbol\textemptyset{"2205}
10872 \DeclareUTFSymbol\texttriangle{"2206}
10873 \DeclareUTFSymbol\textnabla{"2207}
10874 \DeclareUTFSymbol\textin{"2208}
10875 \DeclareUTFSymbol\textnotin{"2209}

```

```

10876 \DeclareUTFSymbol\textsmallin{"220A}
10877 \DeclareUTFSymbol\textni{"220B}
10878 \DeclareUTFSymbol\textnotowner{"220C}
10879 \DeclareUTFSymbol\textsmallowns{"220D}
10880 \DeclareUTFSymbol\textprod{"220F}
10881 \DeclareUTFSymbol\textamalg{"2210}
10882 \DeclareUTFSymbol\textsum{"2211}
10883 \DeclareUTFSymbol\textmp{"2213}
10884 \DeclareUTFSymbol\textdotplus{"2214}
10885 \DeclareUTFSymbol\textDivides{"2215}
10886 \DeclareUTFSymbol\textsetminus{"2216}
10887 \DeclareUTFSymbol\textast{"2217}
10888 \DeclareUTFSymbol\textcirc{"2218}
10889 \DeclareUTFSymbol\textbulletoperator{"2219}
10890 \DeclareUTFSymbol\textpropto{"221D}
10891 \DeclareUTFSymbol\textinfty{"221E}
10892 \DeclareUTFSymbol\textangle{"2220}
10893 \DeclareUTFSymbol\textmeasuredangle{"2221}
10894 \DeclareUTFSymbol\textsphericalangle{"2222}
10895 \DeclareUTFSymbol\textmid{"2223}
10896 \DeclareUTFSymbol\textnmid{"2224}
10897 \DeclareUTFSymbol\textparallel{"2225}
10898 \DeclareUTFSymbol\textnparallel{"2226}
10899 \DeclareUTFSymbol\textwedge{"2227}
10900 \DeclareUTFCommand\textowedge{\textcircled\textwedge}
10901 \DeclareUTFSymbol\textvee{"2228}
10902 \DeclareUTFCommand\textovee{\textcircled\textvee}
10903 \DeclareUTFSymbol\textcap{"2229}
10904 \DeclareUTFSymbol\textcup{"222A}
10905 \DeclareUTFSymbol\textint{"222B}
10906 \DeclareUTFSymbol\textiint{"222C}
10907 \DeclareUTFSymbol\textiiint{"222D}
10908 \DeclareUTFSymbol\textoint{"222E}
10909 \DeclareUTFSymbol\textoiint{"222F}
10910 \DeclareUTFSymbol\textointclockwise{"2232}
10911 \DeclareUTFSymbol\textointctrclockwise{"2233}
10912 \DeclareUTFSymbol\texttherefore{"2234}
10913 \DeclareUTFSymbol\textbecause{"2235}
10914 \DeclareUTFSymbol\textvdotdot{"2236}
10915 \DeclareUTFSymbol\textsquaredots{"2237}
10916 \DeclareUTFSymbol\textdotminus{"2238}
10917 \DeclareUTFSymbol\texteqcolon{"2239}
10918 \DeclareUTFSymbol\textsim{"223C}
10919 \DeclareUTFSymbol\textbacksim{"223D}
10920 \DeclareUTFCommand\textnbacksim{\textlstrikethru\textbacksim}
10921 \DeclareUTFSymbol\textwr{"2240}
10922 \DeclareUTFSymbol\textnsim{"2241}
10923 \DeclareUTFSymbol\texteqsim{"2242}
10924 \DeclareUTFCommand\textneqsim{\textlstrikethru\texteqsim}
10925 \DeclareUTFSymbol\textsimeq{"2243}
10926 \DeclareUTFSymbol\textnsimeq{"2244}
10927 \DeclareUTFSymbol\textcong{"2245}
10928 \DeclareUTFSymbol\textncong{"2247}
10929 \DeclareUTFSymbol\textapprox{"2248}
10930 \DeclareUTFSymbol\textnapprox{"2249}
10931 \DeclareUTFSymbol\textapproxeq{"224A}
10932 \DeclareUTFCommand\textnapproxeq{\textlstrikethru\textapproxeq}
10933 \DeclareUTFSymbol\texttriplesim{"224B}
10934 \DeclareUTFCommand\textntriplesim{\textlstrikethru\texttriplesim}
10935 \DeclareUTFSymbol\textbackcong{"224C}
10936 \DeclareUTFCommand\textnbackcong{\textlstrikethru\textbackcong}
10937 \DeclareUTFSymbol\textasympt{"224D}
10938 \DeclareUTFCommand\textnasympt{\textlstrikethru\textasympt}
10939 \DeclareUTFSymbol\textBumpeq{"224E}
10940 \DeclareUTFCommand\textnBumpeq{\textlstrikethru\textBumpeq}
10941 \DeclareUTFSymbol\textbumpeq{"224F}
10942 \DeclareUTFCommand\textnbumpeq{\textlstrikethru\textbumpeq}

```

```

10943 \DeclareUTFSymbol\textdoteq{"2250}
10944 \DeclareUTFCommand\textndoteq{\textlstrikethru\textdoteq}
10945 \DeclareUTFSymbol\textdoteqdot{"2251}
10946 \DeclareUTFCommand\textnDoteq{\textlstrikethru\textdoteqdot}
10947 \DeclareUTFSymbol\textfallingdoteq{"2252}
10948 \DeclareUTFCommand\textnfallingdoteq{\textlstrikethru\textfallingdoteq}
10949 \DeclareUTFSymbol\textrisingdoteq{"2253}
10950 \DeclareUTFCommand\textnrisingdoteq{\textlstrikethru\textrisingdoteq}
10951 \DeclareUTFSymbol\textcolonequals{"2254}
10952 \DeclareUTFSymbol\textequalscolon{"2255}
10953 \DeclareUTFSymbol\texteqcirc{"2256}
10954 \DeclareUTFCommand\textneqcirc{\textlstrikethru\texteqcirc}
10955 \DeclareUTFSymbol\textcirceq{"2257}
10956 \DeclareUTFCommand\textncirceq{\textlstrikethru\textcirceq}
10957 \DeclareUTFSymbol\texthateq{"2259}
10958 \DeclareUTFCommand\textnhateq{\textlstrikethru\texthateq}
10959 \DeclareUTFSymbol\texttriangleeq{"225C}
10960 \DeclareUTFSymbol\textneq{"2260}
10961 \DeclareUTFSymbol\textne{"2260}
10962 \DeclareUTFSymbol\textequiv{"2261}
10963 \DeclareUTFSymbol\textnequiv{"2262}
10964 \DeclareUTFSymbol\textleq{"2264}
10965 \DeclareUTFSymbol\textle{"2264}
10966 \DeclareUTFSymbol\textgeq{"2265}
10967 \DeclareUTFSymbol\textge{"2265}
10968 \DeclareUTFSymbol\textleqq{"2266}
10969 \DeclareUTFCommand\textnleqq{\textlstrikethru\textleqq}
10970 \DeclareUTFSymbol\textgeqq{"2267}
10971 \DeclareUTFCommand\textngeqq{\textlstrikethru\textgeqq}
10972 \DeclareUTFSymbol\textlneqq{"2268}
10973 \DeclareUTFSymbol\textgneqq{"2269}
10974 \DeclareUTFSymbol\textll{"226A}
10975 \DeclareUTFCommand\textnll{\textlstrikethru\textll}
10976 \DeclareUTFSymbol\textgg{"226B}
10977 \DeclareUTFCommand\textngg{\textlstrikethru\textgg}
10978 \DeclareUTFSymbol\textbetween{"226C}
10979 \DeclareUTFSymbol\textnless{"226E}
10980 \DeclareUTFSymbol\textngtr{"226F}
10981 \DeclareUTFSymbol\textnleq{"2270}
10982 \DeclareUTFSymbol\textngeq{"2271}
10983 \DeclareUTFSymbol\textlessssim{"2272}
10984 \DeclareUTFSymbol\textgtrsim{"2273}
10985 \DeclareUTFSymbol\textnlesssim{"2274}
10986 \DeclareUTFSymbol\textngtrsim{"2275}
10987 \DeclareUTFSymbol\textlessgtr{"2276}
10988 \DeclareUTFSymbol\textgtrless{"2277}
10989 \DeclareUTFSymbol\textngtrless{"2278}
10990 \DeclareUTFSymbol\textnlessgtr{"2279}
10991 \DeclareUTFSymbol\textprec{"227A}
10992 \DeclareUTFSymbol\textsucc{"227B}
10993 \DeclareUTFSymbol\textpreccurlyeq{"227C}
10994 \DeclareUTFSymbol\textsucceedcurlyeq{"227D}
10995 \DeclareUTFSymbol\textprecsim{"227E}
10996 \DeclareUTFCommand\textnprecsim{\textlstrikethru\textprecsim}
10997 \DeclareUTFSymbol\textsuccsim{"227F}
10998 \DeclareUTFCommand\textnsuccsim{\textlstrikethru\textsuccsim}
10999 \DeclareUTFSymbol\textnprec{"2280}
11000 \DeclareUTFSymbol\textnsucc{"2281}
11001 \DeclareUTFSymbol\textsubset{"2282}
11002 \DeclareUTFSymbol\textsupset{"2283}
11003 \DeclareUTFSymbol\textnsubset{"2284}
11004 \DeclareUTFSymbol\textnsupset{"2285}
11005 \DeclareUTFSymbol\textsubseteq{"2286}
11006 \DeclareUTFSymbol\textsupseteq{"2287}
11007 \DeclareUTFSymbol\textnsubseteq{"2288}
11008 \DeclareUTFSymbol\textnsupseteq{"2289}
11009 \DeclareUTFSymbol\textsubsetneq{"228A}

```

```

11010 \DeclareUTFSymbol\textsupsetneq{"228B}
11011 \DeclareUTFSymbol\textcupdot{"228D}
11012 \DeclareUTFSymbol\textcupplus{"228E}
11013 \DeclareUTFSymbol\textsqsubset{"228F}
11014 \DeclareUTFCommand\textnsqsubset{\textlstrikethru\textsqsubset}
11015 \DeclareUTFSymbol\textsqsupset{"2290}
11016 \DeclareUTFCommand\textnsqsupset{\textlstrikethru\textsqsupset}
11017 \DeclareUTFSymbol\textsqsubseq{"2291}
11018 \DeclareUTFCommand\textnsqsubseq{\textlstrikethru\textsqsubseq}
11019 \DeclareUTFSymbol\textsqsupseq{"2292}
11020 \DeclareUTFCommand\textnsqsupseq{\textlstrikethru\textsqsupseq}
11021 \DeclareUTFSymbol\textsqcap{"2293}
11022 \DeclareUTFSymbol\textsqcup{"2294}
11023 \DeclareUTFSymbol\textoplus{"2295}
11024 \DeclareUTFSymbol\textominus{"2296}
11025 \DeclareUTFSymbol\textotimes{"2297}
11026 \DeclareUTFSymbol\textoslash{"2298}
11027 \DeclareUTFSymbol\textodot{"2299}
11028 \DeclareUTFSymbol\textcircledcirc{"229A}
11029 \DeclareUTFSymbol\textcircledast{"229B}
11030 \DeclareUTFSymbol\textcircledash{"229D}
11031 \DeclareUTFSymbol\textboxplus{"229E}
11032 \DeclareUTFSymbol\textboxminus{"229F}
11033 \DeclareUTFSymbol\textboxtimes{"22A0}
11034 \DeclareUTFSymbol\textboxdot{"22A1}
11035 \DeclareUTFSymbol\textvdash{"22A2}
11036 \DeclareUTFSymbol\textdashv{"22A3}
11037 \DeclareUTFCommand\textndashv{\textlstrikethru\textdashv}
11038 \DeclareUTFSymbol\texttop{"22A4}
11039 \DeclareUTFCommand\textndownvdash{\textlstrikethru\texttop}
11040 \DeclareUTFSymbol\textbot{"22A5}
11041 \DeclareUTFCommand\textnupvdash{\textlstrikethru\textbot}
11042 \DeclareUTFSymbol\textvDash{"22A8}
11043 \DeclareUTFSymbol\textVdash{"22A9}
11044 \DeclareUTFSymbol\textVvdash{"22AA}
11045 \DeclareUTFCommand\textnVdash{\textlstrikethru\textVvdash}
11046 \DeclareUTFSymbol\textVDash{"22AB}
11047 \DeclareUTFSymbol\textnvdash{"22AC}
11048 \DeclareUTFSymbol\textnvDash{"22AD}
11049 \DeclareUTFSymbol\textnVdash{"22AE}
11050 \DeclareUTFSymbol\textnVDash{"22AF}
11051 \DeclareUTFSymbol\textlhd{"22B2}
11052 \DeclareUTFSymbol\textrhd{"22B3}
11053 \DeclareUTFSymbol\textunlhd{"22B4}
11054 \DeclareUTFSymbol\textunrhd{"22B5}
11055 \DeclareUTFSymbol\textmultimapdotbothA{"22B6}
11056 \DeclareUTFSymbol\textmultimapdotbothB{"22B7}
11057 \DeclareUTFSymbol\textmultimap{"22B8}
11058 \DeclareUTFSymbol\textveebar{"22BB}
11059 \DeclareUTFSymbol\textbarwedge{"22BC}
11060 \DeclareUTFSymbol\textstar{"22C6}
11061 \DeclareUTFSymbol\textdivideontimes{"22C7}
11062 \DeclareUTFSymbol\textbowtie{"22C8}
11063 \DeclareUTFSymbol\textltimes{"22C9}
11064 \DeclareUTFSymbol\textrtimes{"22CA}
11065 \DeclareUTFSymbol\textleftthreetimes{"22CB}
11066 \DeclareUTFSymbol\textrightthreetimes{"22CC}
11067 \DeclareUTFSymbol\textbacksimeq{"22CD}
11068 \DeclareUTFCommand\textnbacksimeq{\textlstrikethru\textbacksimeq}
11069 \DeclareUTFSymbol\textcurlyvee{"22CE}
11070 \DeclareUTFSymbol\textcurlywedge{"22CF}
11071 \DeclareUTFSymbol\textSubset{"22D0}
11072 \DeclareUTFCommand\textnSubset{\textlstrikethru\textSubset}
11073 \DeclareUTFSymbol\textSupset{"22D1}
11074 \DeclareUTFCommand\textnSupset{\textlstrikethru\textSupset}
11075 \DeclareUTFSymbol\textCap{"22D2}
11076 \DeclareUTFSymbol\textCup{"22D3}

```

```

11077 \DeclareUTFSymbol\textpitchfork{"22D4}
11078 \DeclareUTFSymbol\textlessdot{"22D6}
11079 \DeclareUTFSymbol\textgtrdot{"22D7}
11080 \DeclareUTFSymbol\textl1l1{"22D8}
11081 \DeclareUTFSymbol\textggg{"22D9}
11082 \DeclareUTFSymbol\textlesseqgtr{"22DA}
11083 \DeclareUTFSymbol\textgtreqless{"22DB}
11084 \DeclareUTFSymbol\textcurlyeqprec{"22DE}
11085 \DeclareUTFCommand\textncurlyeqprec{\textlstrikethru\textcurlyeqprec}
11086 \DeclareUTFSymbol\textcurlyeqsucc{"22DF}
11087 \DeclareUTFCommand\textncurlyeqsucc{\textlstrikethru\textcurlyeqsucc}
11088 \DeclareUTFSymbol\textnpreccurlyeq{"22E0}
11089 \DeclareUTFSymbol\textnsucccurlyeq{"22E1}
11090 \DeclareUTFSymbol\textnqsubsetq{"22E2}
11091 \DeclareUTFSymbol\textnqsupsetq{"22E3}
11092 \DeclareUTFSymbol\textsqsubsetneq{"22E4}
11093 \DeclareUTFSymbol\textsqsupsetneq{"22E5}
11094 \DeclareUTFSymbol\textlnsim{"22E6}
11095 \DeclareUTFSymbol\textgnsim{"22E7}
11096 \DeclareUTFSymbol\textprecnsim{"22E8}
11097 \DeclareUTFSymbol\textsuccnsim{"22E9}
11098 \DeclareUTFSymbol\textntriangleleft{"22EA}
11099 \DeclareUTFSymbol\textntriangleright{"22EB}
11100 \DeclareUTFSymbol\textntrianglelefteq{"22EC}
11101 \DeclareUTFSymbol\textntrianglerighteq{"22ED}
11102 \DeclareUTFSymbol\textvdots{"22EE}
11103 \DeclareUTFSymbol\textcdots{"22EF}
11104 \DeclareUTFSymbol\textudots{"22F0}
11105 \DeclareUTFSymbol\textddots{"22F1}
11106 \DeclareUTFSymbol\textbarin{"22F6}
11107 \DeclareUTFSymbol\textdiameter{"2300}
11108 \DeclareUTFSymbol\textbackneg{"2310}
11109 \DeclareUTFSymbol\textwasylounge{"2311}
11110 \DeclareUTFSymbol\textinvbackneg{"2319}
11111 \DeclareUTFSymbol\textclock{"231A}
11112 \DeclareUTFSymbol\textulcorner{"231C}
11113 \DeclareUTFSymbol\texturcorner{"231D}
11114 \DeclareUTFSymbol\textllcorner{"231E}
11115 \DeclareUTFSymbol\textlrcorner{"231F}
11116 \DeclareUTFSymbol\textfrown{"2322}
11117 \DeclareUTFSymbol\textsmile{"2323}
11118 \DeclareUTFSymbol\textKeyboard{"2328}
11119 \DeclareUTFSymbol\textlangle{"2329}
11120 \DeclareUTFSymbol\textrangle{"232A}
11121 \DeclareUTFSymbol\textAPLin{\textA}{"2339}
11122 \DeclareUTFSymbol\textTumbler{"233C}
11123 \DeclareUTFSymbol\textstmaryrdbar{"233D}
11124 \DeclareUTFSymbol\textnotslash{"233F}
11125 \DeclareUTFSymbol\textnotbackslash{"2340}
11126 \DeclareUTFSymbol\textboxbackslash{"2342}
11127 \DeclareUTFSymbol\textAPLleftarrowbox{"2347}
11128 \DeclareUTFSymbol\textAPLrightarrowbox{"2348}
11129 \DeclareUTFSymbol\textAPLuparrowbox{"2350}
11130 \DeclareUTFSymbol\textAPLdownarrowbox{"2357}
11131 \DeclareUTFSymbol\textAPLinput{"235E}
11132 \DeclareUTFSymbol\textRequest{"2370}
11133 \DeclareUTFSymbol\textBeam{"2393}
11134 \DeclareUTFSymbol\texthexagon{"2394}
11135 \DeclareUTFSymbol\textAPLbox{"2395}
11136 \DeclareUTFSymbol\textForwardToIndex{"23ED}
11137 \DeclareUTFSymbol\textRewindToIndex{"23EE}
11138 \DeclareUTFSymbol\textbbslash{"244A}
11139 \DeclareUTFSymbol\textCircledA{"24B6}
11140 \DeclareUTFSymbol\textCleaningF{"24BB}
11141 \DeclareUTFCommand\textCleaningFF{\b\textCleaningF}
11142 \DeclareUTFSymbol\textCleaningP{"24C5}
11143 \DeclareUTFCommand\textCleaningPP{\b\textCleaningP}

```

```

11144 \DeclareUTFSymbol\textCuttingLine{"2504}
11145 \DeclareUTFSymbol\textUParrow{"25B2}
11146 \DeclareUTFSymbol\textbigtriangleup{"25B3}
11147 \DeclareUTFSymbol\textForward{"25B6}
11148 \DeclareUTFSymbol\texttriangleright{"25B7}
11149 \DeclareUTFSymbol\textRHD{"25BA}
11150 \DeclareUTFSymbol\textDOWNarrow{"25BC}
11151 \DeclareUTFSymbol\textbigtriangledown{"25BD}
11152 \DeclareUTFSymbol\textRewind{"25C0}
11153 \DeclareUTFSymbol\texttriangleleft{"25C1}
11154 \DeclareUTFSymbol\textLHD{"25C4}
11155 \DeclareUTFSymbol\textdiamond{"25C7}
11156 \DeclareUTFSymbol\textlozenge{"25CA}
11157 \DeclareUTFSymbol\textLEFTCIRCLE{"25D6}
11158 \DeclareUTFSymbol\textRIGHTCIRCLE{"25D7}
11159 \DeclareUTFSymbol\textboxbar{"25EB}
11160 \DeclareUTFSymbol\textCloud{"2601}
11161 \DeclareUTFSymbol\textFiveStar{"2605}
11162 \DeclareUTFSymbol\textFiveStarOpen{"2606}
11163 \DeclareUTFSymbol\textPhone{"260E}
11164 \DeclareUTFSymbol\textboxempty{"2610}
11165 \DeclareUTFSymbol\textCheckedbox{"2611}
11166 \DeclareUTFSymbol\textCrossedbox{"2612}
11167 \DeclareUTFSymbol\textCoffeecup{"2615}
11168 \DeclareUTFSymbol\textHandCuffLeft{"261A}
11169 \DeclareUTFSymbol\textHandCuffRight{"261B}
11170 \DeclareUTFSymbol\textHandLeft{"261C}
11171 \DeclareUTFSymbol\textHandRight{"261E}
11172 \DeclareUTFSymbol\textRadioactivity{"2622}
11173 \DeclareUTFSymbol\textBiohazard{"2623}
11174 \DeclareUTFSymbol\textAnkh{"2625}
11175 \DeclareUTFSymbol\textYinYang{"262F}
11176 \DeclareUTFSymbol\textfrownie{"2639}
11177 \DeclareUTFSymbol\textsmiley{"263A}
11178 \DeclareUTFSymbol\textblacksmiley{"263B}
11179 \DeclareUTFSymbol\textsun{"263C}
11180 \DeclareUTFSymbol\textleftmoon{"263D}
11181 \DeclareUTFSymbol\textrightmoon{"263E}
11182 \DeclareUTFSymbol\textmercury{"263F}
11183 \DeclareUTFSymbol\textPUfemale{"2640}
11184 \DeclareUTFSymbol\textearth{"2641}
11185 \DeclareUTFSymbol\textmale{"2642}
11186 \DeclareUTFSymbol\textjupiter{"2643}
11187 \DeclareUTFSymbol\textsaturn{"2644}
11188 \DeclareUTFSymbol\texturanus{"2645}
11189 \DeclareUTFSymbol\textneptune{"2646}
11190 \DeclareUTFSymbol\textpluto{"2647}
11191 \DeclareUTFSymbol\textaries{"2648}
11192 \DeclareUTFSymbol\texttaurus{"2649}
11193 \DeclareUTFSymbol\textgemini{"264A}
11194 \DeclareUTFSymbol\textcancer{"264B}
11195 \DeclareUTFSymbol\textleo{"264C}
11196 \DeclareUTFSymbol\textvirgo{"264D}
11197 \DeclareUTFSymbol\textlibra{"264E}
11198 \DeclareUTFSymbol\textscorpio{"264F}
11199 \DeclareUTFSymbol\textsagittarius{"2650}
11200 \DeclareUTFSymbol\textcapricornus{"2651}
11201 \DeclareUTFSymbol\textaquarius{"2652}
11202 \DeclareUTFSymbol\textpisces{"2653}
11203 \DeclareUTFSymbol\textspadesuitblack{"2660}
11204 \DeclareUTFSymbol\textheartsuitwhite{"2661}
11205 \DeclareUTFSymbol\textdiamondsuitwhite{"2662}
11206 \DeclareUTFSymbol\textclubsuitblack{"2663}
11207 \DeclareUTFSymbol\textspadesuitwhite{"2664}
11208 \DeclareUTFSymbol\textheartsuitblack{"2665}
11209 \DeclareUTFSymbol\textdiamondsuitblack{"2666}
11210 \DeclareUTFSymbol\textclubsuitwhite{"2667}

```

```

11211 \DeclareUTFSymbol\textquaternnote{"2669}
11212 \DeclareUTFSymbol\texttwoonotes{"266B}
11213 \DeclareUTFSymbol\textsixteenthnote{"266C}
11214 \DeclareUTFSymbol\textflat{"266D}
11215 \DeclareUTFSymbol\textnatural{"266E}
11216 \DeclareUTFSymbol\textsharp{"266F}
11217 \DeclareUTFSymbol\textrecycle{"2672}
11218 \DeclareUTFSymbol\textWheelchair{"267F}
11219 \DeclareUTFSymbol\textFlag{"2691}
11220 \DeclareUTFSymbol\textMineSign{"2692}
11221 \DeclareUTFSymbol\textdsilitary{"2694}
11222 \DeclareUTFSymbol\textdsmedical{"2695}
11223 \DeclareUTFSymbol\textdsjuridical{"2696}
11224 \DeclareUTFSymbol\textdschemical{"2697}
11225 \DeclareUTFSymbol\textdsbiological{"2698}
11226 \DeclareUTFSymbol\textdscommercial{"269A}
11227 \DeclareUTFSymbol\textmanstar{"269D}
11228 \DeclareUTFSymbol\textdanger{"26A0}
11229 \DeclareUTFSymbol\textFemaleFemale{"26A2}
11230 \DeclareUTFSymbol\textMaleMale{"26A3}
11231 \DeclareUTFSymbol\textFemaleMale{"26A4}
11232 \DeclareUTFSymbol\textHermaphrodite{"26A5}
11233 \DeclareUTFSymbol\textNeutral{"26AA}
11234 \DeclareUTFSymbol\textPUuncrfemale{"26B2}
11235 \DeclareUTFSymbol\texthexstar{"26B9}
11236 \DeclareUTFSymbol\textSoccerBall{"26BD}
11237 \DeclareUTFSymbol\textSunCloud{"26C5}
11238 \DeclareUTFSymbol\textRain{"26C6}
11239 \DeclareUTFSymbol\textnoway{"26D4}
11240 \DeclareUTFSymbol\textMountain{"26F0}
11241 \DeclareUTFSymbol\textTent{"26FA}
11242 \DeclareUTFSymbol\textScissorRightBrokenBottom{"2701}
11243 \DeclareUTFSymbol\textScissorRight{"2702}
11244 \DeclareUTFSymbol\textScissorRightBrokenTop{"2703}
11245 \DeclareUTFSymbol\textScissorHollowRight{"2704}
11246 \DeclareUTFSymbol\textPhoneHandset{"2706}
11247 \DeclareUTFSymbol\textTape{"2707}
11248 \DeclareUTFSymbol\textPlane{"2708}
11249 \DeclareUTFSymbol\textEnvelope{"2709}
11250 \DeclareUTFSymbol\textPeace{"270C}
11251 \DeclareUTFSymbol\textWritingHand{"270D}
11252 \DeclareUTFSymbol\textPencilRightDown{"270E}
11253 \DeclareUTFSymbol\textPencilRight{"270F}
11254 \DeclareUTFSymbol\textPencilRightUp{"2710}
11255 \DeclareUTFSymbol\textNibRight{"2711}
11256 \DeclareUTFSymbol\textNibSolidRight{"2712}
11257 \DeclareUTFSymbol\textCheckmark{"2713}
11258 \DeclareUTFSymbol\textCheckmarkBold{"2714}
11259 \DeclareUTFSymbol\textXSolid{"2715}
11260 \DeclareUTFSymbol\textXSolidBold{"2716}
11261 \DeclareUTFSymbol\textXSolidBrush{"2717}
11262 \DeclareUTFSymbol\textPlusOutline{"2719}
11263 \DeclareUTFSymbol\textPlus{"271A}
11264 \DeclareUTFSymbol\textPlusThinCenterOpen{"271B}
11265 \DeclareUTFSymbol\textPlusCenterOpen{"271C}
11266 \DeclareUTFSymbol\textCross{"271D}
11267 \DeclareUTFSymbol\textCrossOpenShadow{"271E}
11268 \DeclareUTFSymbol\textCrossOutline{"271F}
11269 \DeclareUTFSymbol\textCrossMaltese{"2720}
11270 \DeclareUTFSymbol\textDavidStar{"2721}
11271 \DeclareUTFSymbol\textFourAsterisk{"2722}
11272 \DeclareUTFSymbol\textJackStar{"2723}
11273 \DeclareUTFSymbol\textJackStarBold{"2724}
11274 \DeclareUTFSymbol\textClowerTips{"2725}
11275 \DeclareUTFSymbol\textFourStar{"2726}
11276 \DeclareUTFSymbol\textFourStarOpen{"2727}
11277 \DeclareUTFSymbol\textFiveStarOpenCircled{"272A}

```

```

11278 \DeclareUTFSymbol\textFiveStarCenterOpen{"272B}
11279 \DeclareUTFSymbol\textFiveStarOpenDotted{"272C}
11280 \DeclareUTFSymbol\textFiveStarOutline{"272D}
11281 \DeclareUTFSymbol\textFiveStarOutlineHeavy{"272E}
11282 \DeclareUTFSymbol\textFiveStarConvex{"272F}
11283 \DeclareUTFSymbol\textFiveStarShadow{"2730}
11284 \DeclareUTFSymbol\textAsteriskBold{"2731}
11285 \DeclareUTFSymbol\textAsteriskCenterOpen{"2732}
11286 \DeclareUTFSymbol\textEightStarTaper{"2734}
11287 \DeclareUTFSymbol\textEightStarConvex{"2735}
11288 \DeclareUTFSymbol\textSixStar{"2736}
11289 \DeclareUTFSymbol\textEightStar{"2737}
11290 \DeclareUTFSymbol\textEightStarBold{"2738}
11291 \DeclareUTFSymbol\textTwelveStar{"2739}
11292 \DeclareUTFSymbol\textSixteenStarLight{"273A}
11293 \DeclareUTFSymbol\textSixFlowerPetalRemoved{"273B}
11294 \DeclareUTFSymbol\textSixFlowerOpenCenter{"273C}
11295 \DeclareUTFSymbol\textAsterisk{"273D}
11296 \DeclareUTFSymbol\textSixFlowerAlternate{"273E}
11297 \DeclareUTFSymbol\textFiveFlowerPetal{"273F}
11298 \DeclareUTFSymbol\textFiveFlowerOpen{"2740}
11299 \DeclareUTFSymbol\textEightFlowerPetal{"2741}
11300 \DeclareUTFSymbol\textSunshineOpenCircled{"2742}
11301 \DeclareUTFSymbol\textSixFlowerAltPetal{"2743}
11302 \DeclareUTFSymbol\textSnowflakeChevron{"2744}
11303 \DeclareUTFSymbol\textSnowflake{"2745}
11304 \DeclareUTFSymbol\textSnowflakeChevronBold{"2746}
11305 \DeclareUTFSymbol\textSparkle{"2747}
11306 \DeclareUTFSymbol\textSparkleBold{"2748}
11307 \DeclareUTFSymbol\textAsteriskRoundedEnds{"2749}
11308 \DeclareUTFSymbol\textEightFlowerPetalRemoved{"274A}
11309 \DeclareUTFSymbol\textEightAsterisk{"274B}
11310 \DeclareUTFSymbol\textCircleShadow{"274D}
11311 \DeclareUTFSymbol\textSquareShadowBottomRight{"274F}
11312 \DeclareUTFSymbol\textSquareTopRight{"2750}
11313 \DeclareUTFSymbol\textSquareCastShadowBottomRight{"2751}
11314 \DeclareUTFSymbol\textSquareCastShadowTopRight{"2752}
11315 \DeclareUTFSymbol\textDiamondSolid{"2756}
11316 \DeclareUTFSymbol\textRectangleThin{"2758}
11317 \DeclareUTFSymbol\textRectangle{"2759}
11318 \DeclareUTFSymbol\textRectangleBold{"275A}
11319 \DeclareUTFSymbol\textperp{"27C2}
11320 \DeclareUTFCommand\textnotperp{\textlstrikethru\textperp}
11321 \DeclareUTFSymbol\textveedot{"27C7}
11322 \DeclareUTFSymbol\textwedgedot{"27D1}
11323 \DeclareUTFSymbol\textleftspoon{"27DC}
11324 \DeclareUTFSymbol\textlbrackdbl{"27E6}
11325 \DeclareUTFSymbol\textrrackdbl{"27E7}
11326 \DeclareUTFSymbol\textcirclearrowleft{"27F2}
11327 \DeclareUTFSymbol\textcirclearrowright{"27F3}
11328 \DeclareUTFSymbol\textlongleftarrow{"27F5}
11329 \DeclareUTFSymbol\textlongrightarrow{"27F6}
11330 \DeclareUTFSymbol\textlongleftrightarrow{"27F7}
11331 \DeclareUTFSymbol\textLongleftarrow{"27F8}
11332 \DeclareUTFSymbol\textLongrightarrow{"27F9}
11333 \DeclareUTFSymbol\textLongleftrightarrow{"27FA}
11334 \DeclareUTFSymbol\textlongmapsto{"27FC}
11335 \DeclareUTFSymbol\textLongmapsfrom{"27FD}
11336 \DeclareUTFSymbol\textLongmapsto{"27FE}
11337 \DeclareUTFSymbol\textnwsearrow{"2921}
11338 \DeclareUTFSymbol\textneswarrow{"2922}
11339 \DeclareUTFSymbol\textlhooknarrow{"2923}
11340 \DeclareUTFSymbol\textrhoonknearrow{"2924}
11341 \DeclareUTFSymbol\textlhooksearrow{"2925}
11342 \DeclareUTFSymbol\textrhoonksearrow{"2926}
11343 \DeclareUTFSymbol\textleadsto{"2933}
11344 \DeclareUTFSymbol\textrcurvearrowne{"2934}

```

```

11345 \DeclareUTFSymbol\textlcurvearrowse{"2935}
11346 \DeclareUTFSymbol\textlcurvearrowsw{"2936}
11347 \DeclareUTFSymbol\textrcurvearrowse{"2937}
11348 \DeclareUTFSymbol\textlcurvearrowdown{"2938}
11349 \DeclareUTFSymbol\textrcurvearrowdown{"2939}
11350 \DeclareUTFSymbol\textrcurvearrowleft{"293A}
11351 \DeclareUTFSymbol\textrcurvearrowright{"293B}
11352 \DeclareUTFSymbol\textlefttrightharpoon{"294A}
11353 \DeclareUTFSymbol\textrightleftharpoon{"294B}
11354 \DeclareUTFSymbol\textupdownharpoonrightleft{"294C}
11355 \DeclareUTFSymbol\textupdownharpoonleftright{"294D}
11356 \DeclareUTFSymbol\textleftleftharpoons{"2962}
11357 \DeclareUTFSymbol\textupupharpoons{"2963}
11358 \DeclareUTFSymbol\textrighttrightharpoons{"2964}
11359 \DeclareUTFSymbol\textdowndownharpoons{"2965}
11360 \DeclareUTFSymbol\textleftbarharpoon{"296A}
11361 \DeclareUTFSymbol\textbarleftharpoon{"296B}
11362 \DeclareUTFSymbol\textrightbarharpoon{"296C}
11363 \DeclareUTFSymbol\textbarrightharpoon{"296D}
11364 \DeclareUTFSymbol\textupdownharpoons{"296E}
11365 \DeclareUTFSymbol\textdownupharpoons{"296F}
11366 \DeclareUTFSymbol\textllparenthesis{"2987}
11367 \DeclareUTFSymbol\textrrparenthesis{"2988}
11368 \DeclareUTFSymbol\textinvdiameter{"29B0}
11369 \DeclareUTFSymbol\textobar{"29B6}
11370 \DeclareUTFSymbol\textobslash{"29B8}
11371 \DeclareUTFSymbol\textobot{"29BA}
11372 \DeclareUTFSymbol\textNoChemicalCleaning{"29BB}
11373 \DeclareUTFSymbol\textolessthan{"29C0}
11374 \DeclareUTFSymbol\textogreaterthan{"29C1}
11375 \DeclareUTFSymbol\textboxslash{"29C4}
11376 \DeclareUTFSymbol\textboxbslash{"29C5}
11377 \DeclareUTFSymbol\textboxast{"29C6}
11378 \DeclareUTFSymbol\textboxcircle{"29C7}
11379 \DeclareUTFSymbol\textboxbox{"29C8}
11380 \DeclareUTFSymbol\textValve{"29D3}
11381 \DeclareUTFSymbol\textmultimapboth{"29DF}
11382 \DeclareUTFSymbol\textshuffle{"29E2}
11383 \DeclareUTFSymbol\textuplus{"2A04}
11384 \DeclareUTFSymbol\textbigdoublewedge{"2A07}
11385 \DeclareUTFSymbol\textbigdoublevee{"2A08}
11386 \DeclareUTFSymbol\textJoin{"2A1D}
11387 \DeclareUTFSymbol\textfatsemi{"2A1F}
11388 \DeclareUTFSymbol\textcircplus{"2A22}
11389 \DeclareUTFSymbol\textminusdot{"2A2A}
11390 \DeclareUTFSymbol\textdottimes{"2A30}
11391 \DeclareUTFSymbol\textdttimes{"2A32}
11392 \DeclareUTFSymbol\textodiv{"2A38}
11393 \DeclareUTFSymbol\textinvneg{"2A3C}
11394 \DeclareUTFSymbol\textsqdoublecap{"2A4E}
11395 \DeclareUTFSymbol\textcapdot{"2A40}
11396 \DeclareUTFSymbol\textsqdoublecup{"2A4F}
11397 \DeclareUTFSymbol\textdoublewedge{"2A55}
11398 \DeclareUTFSymbol\textdoublevee{"2A56}
11399 \DeclareUTFSymbol\textdoublebarwedge{"2A5E}
11400 \DeclareUTFSymbol\textveedoublebar{"2A63}
11401 \DeclareUTFSymbol\texteqdot{"2A66}
11402 \DeclareUTFCommand\textneqdot{\textlstrikethru\texteqdot}
11403 \DeclareUTFSymbol\textcoloncolonequals{"2A74}
11404 \DeclareUTFSymbol\textleqslant{"2A7D}
11405 \DeclareUTFCommand\textnleqslant{\textlstrikethru\textleqslant}
11406 \DeclareUTFSymbol\textgeqslant{"2A7E}
11407 \DeclareUTFCommand\textngeqslant{\textlstrikethru\textgeqslant}
11408 \DeclareUTFSymbol\textlessapprox{"2A85}
11409 \DeclareUTFCommand\textnlessapprox{\textlstrikethru\textlessapprox}
11410 \DeclareUTFSymbol\textgtrapprox{"2A86}
11411 \DeclareUTFCommand\textngtrapprox{\textlstrikethru\textgtrapprox}

```

```

11412 \DeclareUTFSymbol\textlneq{"2A87}
11413 \DeclareUTFSymbol\textgneq{"2A88}
11414 \DeclareUTFSymbol\textlnapprox{"2A89}
11415 \DeclareUTFSymbol\textgnapprox{"2A8A}
11416 \DeclareUTFSymbol\textlesseqqgtr{"2A8B}
11417 \DeclareUTFSymbol\textgtreqqless{"2A8C}
11418 \DeclareUTFSymbol\texteqslantless{"2A95}
11419 \DeclareUTFSymbol\texteqslantgtr{"2A96}
11420 \DeclareUTFSymbol\textleftslice{"2AA6}
11421 \DeclareUTFSymbol\textright slice{"2AA7}
11422 \DeclareUTFSymbol\textpreceq{"2AAF}
11423 \DeclareUTFCommand\textnpreceq{\textlstrikethru\textpreceq}
11424 \DeclareUTFSymbol\textsucceq{"2AB0}
11425 \DeclareUTFCommand\textnsucceq{\textlstrikethru\textsucceq}
11426 \DeclareUTFSymbol\textprecneq{"2AB1}
11427 \DeclareUTFSymbol\textsucneq{"2AB2}
11428 \DeclareUTFSymbol\textpreceqq{"2AB3}
11429 \DeclareUTFCommand\textnpreceqq{\textlstrikethru\textpreceqq}
11430 \DeclareUTFSymbol\textsucceqq{"2AB4}
11431 \DeclareUTFCommand\textnsucceqq{\textlstrikethru\textsucceqq}
11432 \DeclareUTFSymbol\textprecneqq{"2AB5}
11433 \DeclareUTFSymbol\textsucneqq{"2AB6}
11434 \DeclareUTFSymbol\textprecapprox{"2AB7}
11435 \DeclareUTFCommand\textnprecapprox{\textlstrikethru\textprecapprox}
11436 \DeclareUTFSymbol\textsuccapprox{"2AB8}
11437 \DeclareUTFCommand\textnsuccapprox{\textlstrikethru\textsuccapprox}
11438 \DeclareUTFSymbol\textprecnapprox{"2AB9}
11439 \DeclareUTFSymbol\textsuccnapprox{"2ABA}
11440 \DeclareUTFSymbol\textsubseteqq{"2AC5}
11441 \DeclareUTFCommand\textnsubseteqq{\textlstrikethru\textsubseteqq}
11442 \DeclareUTFSymbol\textsupseteqq{"2AC6}
11443 \DeclareUTFCommand\textnsupseteqq{\textlstrikethru\textsupseteqq}
11444 \DeclareUTFSymbol\textdashV{"2AE3}
11445 \DeclareUTFCommand\textndashV{\textlstrikethru\textdashV}
11446 \DeclareUTFSymbol\textDashv{"2AE4}
11447 \DeclareUTFCommand\textnDashv{\textlstrikethru\textDashv}
11448 \DeclareUTFSymbol\textDashV{"2AE5}
11449 \DeclareUTFCommand\textnDashV{\textlstrikethru\textDashV}
11450 \DeclareUTFSymbol\textdownmodels{"2AEA}
11451 \DeclareUTFCommand\textndownmodels{\textlstrikethru\textdownmodels}
11452 \DeclareUTFSymbol\textupmodels{"2AEB}
11453 \DeclareUTFCommand\textnupmodels{\textlstrikethru\textupmodels}
11454 \DeclareUTFSymbol\textupspoon{"2AEF}
11455 \DeclareUTFSymbol\textinterleave{"2AF4}
11456 \DeclareUTFSymbol\textsslash{"2AFD}
11457 \DeclareUTFSymbol\textpentagon{"2B20}
11458 \DeclareUTFSymbol\textvarhexagon{"2B21}
11459 \DeclareUTFSymbol\textjinferior{"2C7C}
11460 \DeclareUTFSymbol\textslashdiv{"2E13}
11461 \DeclareUTFSymbol\textinterrobangdown{"2E18}
11462 \DeclareUTFSymbol\textfivedots{"2E2D}
11463 \DeclareUTFSymbol\textPUheng{"A727}
11464 \DeclareUTFSymbol\textPULhookfour{"A72C}
11465 \DeclareUTFSymbol\textPUscf{"A730}
11466 \DeclareUTFSymbol\textPUaolig{"A735}
11467 \DeclareUTFSymbol\textoo{"A74F}
11468 \DeclareUTFSymbol\textcircumlow{"A788}
11469 \DeclareUTFSymbol\textfi{"FB01}
11470 \DeclareUTFSymbol\textfl{"FB02}
11471 \DeclareUTFSymbol\textGaPa{"1D13B}
11472 \DeclareUTFSymbol\textHaPa{"1D13C}
11473 \DeclareUTFSymbol\textViPa{"1D13D}
11474 \DeclareUTFSymbol\textAcPa{"1D13E}
11475 \DeclareUTFSymbol\textSePa{"1D13F}
11476 \DeclareUTFSymbol\textZwPa{"1D140}
11477 \DeclareUTFSymbol\textfullnote{"1D15D}
11478 \DeclareUTFSymbol\texthalfnote{"1D15E}

```

```

11479 \DeclareUTFSymbol\textVier{"1D15F}
11480 \DeclareUTFSymbol\textAcht{"1D160}
11481 \DeclareUTFSymbol\textSech{"1D161}
11482 \DeclareUTFSymbol\textZwdr{"1D162}
11483 \DeclareUTFSymbol\textMundus{"1F30D}
11484 \DeclareUTFSymbol\textMoon{"1F319}
11485 \DeclareUTFSymbol\textManFace{"1F468}
11486 \DeclareUTFSymbol\textWomanFace{"1F469}
11487 \DeclareUTFSymbol\textFax{"1F4E0}
11488 \DeclareUTFSymbol\textFire{"1F525}
11489 \DeclareUTFSymbol\textBicycle{"1F6B2}
11490 \DeclareUTFSymbol\textGentsroom{"1F6B9}
11491 \DeclareUTFSymbol\textLadiesroom{"1F6BA}
11492 \DeclareUTFCommand\textcopyleft{\textcircled\textrevc}
11493 \DeclareUTFCommand\textccsa{\textcircled\textcirclearrowleft}
11494 \DeclareUTFSymbol\textglqq{"201E}
11495 \DeclareUTFSymbol\textgrqq{"201C}
11496 \DeclareUTFSymbol\textglq{"201A}
11497 \DeclareUTFSymbol\textgrq{"2018}
11498 \DeclareUTFSymbol\textflqq{"00AB}
11499 \DeclareUTFSymbol\textfrqq{"00BB}
11500 \DeclareUTFSymbol\textflq{"2039}
11501 \DeclareUTFSymbol\textfrq{"203A}
11502 \DeclareUTFSymbol\textneg{"00AC}
11503 \DeclareUTFSymbol\textcdot{"00B7}

11504 </xunextra>

11505 <@=@=xeCJK>

```

5.22 xeCJK.cfg

```
11506 <*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。

```
11507
```

```
11508 </config>
```

版本历史

v3.1.0	(2012/11/13 – 2012/11/21)	<code>\xeCJK@family</code> : 不将参数完全展开。	148
General: 放弃对 <code>\outer</code> 宏的特殊处理。	1	<code>_xeCJK_check_single_space</code> : NN: 使用	
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。	137	<code>\xeCJK_if_CJK_class</code> : NTF 来代替 <code>\int_case</code> : nnn 判断是否是 CJK 字符类。	99
改用 <code>indentfirst</code> 宏包处理缩进的问题。	144	<code>_xeCJK_family_unknown_warning</code> : n: 在没有定义任何 CJK 字体的情况下, 不再重复给出字体没有定义的警告。	129
取消 <code>\cprotect</code> 的外部宏限制。	160	v3.2.0	(2013/04/14 – 2013/05/22)
删除多余的 <code>default-itcorr</code> 结点。	49	General: 增加 IVS 字符类用于处理异体字选择符。	29
使用 <code>xtemplate</code> 宏包的机制来组织标点符号的处理。	107	增加 Verb 选项。	137
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。	142	<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。	130
<code>\xeCJK@fix@penalty</code> : 采用通过不修改原语 <code>\</code> 的方式对修复倾斜校正。	148	<code>_xeCJK_Boundary_and_FullLeft_glue</code> : N: 当全角左标点前面是 <code>hlist</code> , <code>none</code> , <code>glue</code> 和 <code>penalty</code> 等节点时, 压缩其左空白。	90
<code>_xeCJK_fallback_loop</code> : nnNN: 调整备用字体的循环方式。	119	<code>\l_xeCJK_family_tl</code> : 不将其初始化为	
<code>\xeCJK_glyph_if_exist</code> : N: 改进 <code>fontspec</code> 宏包中定义的 <code>\font_glyph_if_exist</code> : NnTF。	25	<code>\CJKfamilydefault</code> 。	129
<code>\xeCJK_hook_for_ulem</code> : : 简化对 <code>ulem</code> 宏包的兼容补丁。	162	<code>\xeCJK_FullLeft_and_Default</code> : : 修正 <code>xeCJK</code> 使西文在部分情况下无法断词的问题。	89
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑 <code>\spaceskip</code> 不为零的情况。	25	<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑到 <code>\spacefactor</code> 和 <code>\xspaceskip</code> 的情况。	25
<code>_xeCJK_switch_font</code> : nn: 改进定义, 加快切换速度。	126	v3.2.1	(2013/05/29)
<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。	138	General: 调整 Verb 选项: 在命令 <code>\verb</code> 里使用时, 不破坏标点禁则, 增加值 <code>env+</code> 。	137
v3.1.1	(2012/12/02 – 2012/12/13)	v3.2.2	(2013/05/30 – 2013/06/04)
General: 不再依赖 <code>xpatch</code> 宏包。	1	General: 修正某些重音不能正确显示的问题。	1
对于与 <code>xltxtra</code> 的冲突给出错误警告。	148	增加小宏包 <code>xeCJK-listings</code> , 用于支持 <code>listings</code> 宏包。	181
增加 <code>NewLineCS</code> 和 <code>EnvCS</code> 选项。	100	<code>_xeCJK_ulem_CJK_and_FullRight_glue</code> : N: 修正下划线不能跳过全角右标点的问题。	170
增加小宏包 <code>xeCJKintef</code> , 用于处理下划线的问题。	162	v3.2.3	(2013/06/04 – 2013/06/11)
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。	95	General: 不再改变 CJK 字符类的 <code>\catcode</code> 。	37
<code>InlineEnv</code> : 改变行内环境的设置方式, 从而使用 <code>\str_case_x</code> : nnn 代替原来的 <code>\clist_if_in</code> : NnTF 来判断是否是行内环境。	101	根据 <code>X_gTeX</code> 的脚本重新整理全角标点符号。	31
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。	99	解决 <code>CheckSingle</code> 选项与 <code>tablists</code> 宏包的冲突。	100
<code>\xeCJK@family</code> : 修改主要 CJK 字体族的自动更新方式。	148	提供四个 <code>TECkit</code> 映射文件用于句号转换和简繁互换。	1
<code>_xeCJK_check_single_aux</code> : nNNw: 改进定义, 减少使用 <code>peek</code> 函数的次数。	99	完善对 <code>listings</code> 宏包的支持。	181
<code>_xeCJK_check_single_space</code> : NN: <code>CheckSingle</code> 支持段末“汉字 + 汉字 + 空格 + 汉字/标点”的形式。	99	<code>_xeCJK_listings_initial_hook</code> : : 解决 <code>listings</code> 环境中代码行号输出不正确的问题, 并解决在其中跨页时对页眉和页脚的影响。	181
<code>\xeCJK_hook_for_ulem</code> : : 完全处理下划线里的标点符号的有关问题。	162	<code>_xeCJK_listings_process_Default</code> : nN: 在 <code>listings</code> 环境中对 <code>\charcode</code> 大于 255 的字符根据其 <code>\catcode</code> 区分 <code>letter</code> 和 <code>other</code> 。	183
<code>\xeCJK_peek_catcode_ignore_spaces</code> : NTF: 新增有省略空格标识的 <code>peek</code> 函数。	27	<code>_xeCJK_restore_shipout_CJKsymbol</code> : : 解决 <code>\CJKunderdot</code> 跨页使用时影响到页眉页脚的问题。	180
<code>\xeCJK_save_class</code> : nn: 使用 <code>\xeCJK_save_class</code> : nn 保存 <code>X_gTeX</code> 预定义的字符类别。	29	<code>_xeCJK_ulem_FullLeft_and_CJK</code> : : 修正全角左标点后下划线与 <code>\CJKunderdot</code> 连用时结果不正常的问题。	171
<code>\xeCJK_set_char_class</code> : nnn: 在文档中设置字符类别时不重复设置 <code>\catcode</code> 。	37	<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKOffVerbAddon</code> 用于局部取消 <code>\xeCJKOffVerbAddon</code> 的影响; 并解决跨页使用时影响到页眉页脚的问题。	138
<code>_xeCJK_set_char_class_eq</code> : nn: 交换参数的顺序。	37	v3.2.4	(2013/06/23 – 2013/07/06)
<code>_xeCJK_set_verb_exspace</code> : : 调整间距的计算方法。	140	General: 不再使用 <code>CJKnumber</code> 选项, 可以在 <code>xeCJK</code> 之后直接使用 <code>CJKnumb</code> 宏包得到中文数字。	161
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。	97	改进获取分区字体属性的办法。	121
v3.1.2	(2012/12/27 – 2013/01/01)	解决使用 <code>CheckSingle</code> 时, 某些 <code>\CJKglue</code> 不能被正确加入的问题。	100
General: 解决在下划线状态下使用 <code>\makebox</code> 时的错误。	167	尽量移除用作判断标志的 <code>\kern</code> 。	49
修正非 <code>\UTFencname</code> 编码下面 <code>xunicode</code> 重定义的 <code>\nobreakspace</code> 会失效的问题。	148	内部调整分区字体的设置方法。	121
修正重定义 <code>\CJKfamilydefault</code> 无效的问题, 恢复容错能力。	133		

使 listings 的 breaklines 选项对 CJK 字符类可用, 并保持标点符号的禁则。	184	修正 unicode-letters.tex 中谚文符号 \catcode 不准的问题。	37
使用 AllowBreakBetweenPuncts 时, 相应标点符号仍能与边界对齐。	88	\Url@MathSetup: 使通过 \UrlFont 等命令设置的 CJK 字体生效。	145
修正 xeCJKfntef 与 natbib 等的冲突。	162	_xeCJK_check_single_aux:n: 与 \CJKspace 兼容。	99
遵循 L ^A T _E X3 变量需要预先声明的原则。	1	_xeCJK_punct_glue:NN: 标点符号左/右空白的伸展值不超过原始边界, 收缩值不小于另一侧边界。	87
\addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。	132	\xeCJK_set_mathfont:: 将 CJK 字符的数学归类由 7 改为 0, 解决汉字路径的问题。	135
CJKfilltwosides: 改用 minipage 和 L ^A T _E X 表格 (tabular) 来实现。	180	v3.2.8 (2013/11/16 – 2013/12/05)	
_xeCJK_Boundary_and_FullLeft_glue:N: 细化边界与全角左标点之间是否压缩空白的判断。	90	General: 启用 xunicode 中的带圈数字和字母设置。	200
_xeCJK_fallback_loop:nnNN: 使 \CJKfamilydefault 的 FallBack 设置全局可用。	119	\DeclareUTFmathsymbols: 修正 \UseMathAsText 的功能, 恢复 \hbar 和增加以 text 打头的文本符号命令。	189
_xeCJK_set_verb_exspace:: 当计算得出的间距为负时, 缩小 CJK 字体。	140	_xeCJK_nobreak_skip:: 禁止在 \verb 中断行。	138
\xeCJK_tl_remove_outer_braces:n: 去掉外层分组括号时, 移除空格, 避免死循环。	24	\xeCJKVerbAddon: 增加是否是等宽字体的判断。	138
\xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。	27	v3.2.9 (2013/12/07 – 2013/12/08)	
v3.2.5 (2013/07/10 – 2013/07/25)		General: 文档部分增加 xunicode 定义的符号表。	188
General: 恢复 \nobreakspace 的原始定义。	148	增加 xunicode-extra.def 中, 用于加入 puenc.def 中的符号定义。	203
解决 fixltx2e 和 amsthm 的冲突。	148	\DeclareEncodedCompositeAccents: 修正 xunicode 中的错误定义。	193
修正 CJK 和 NormalSpace 字符类之间因为边界造成的间距不正确的问题。	45	\c_xeCJK_middle_dot_prop: 完整处理 encguide.pdf 的编码符号表中, 与旧编码的 U+00B7 冲突。	152
增加小宏包 xunicode-addon, 为 xunicode 提供判断字符是否存在的功能。	188	v3.2.10 (2014/02/20 – 2014/03/01)	
\@setupverbvisiblespace: 可视空格考虑传统 T _E X 字体的情况。	141	\CJKaddEncHook: 使用 CJKnumb 时, 让 \Unicode 有定义。	161
Verb: 微调定义。	137	\DeclareUTFDoubleEncodedAccent: 改进 \t 等的定义方式。	193
_xeCJK_Boundary_and_FullLeft_glue:N: 细化全角左标点是否位于段首的判断。	90	\DeclareUTFDoubleEncodedSymbol: 改进 \sliding 等的定义方式。	193
增加对 enumitem 宏包修改的 \item 的判断。	90	\DeclareUTFTIPACCommand: 检查 \t 和 \sliding 的参数是否以 \textipa 开头。	198
_xeCJK_boundary_register_text_command:NN: 解决汉字后紧跟 \(\dots\) 形式的行内数学公式时, 不能加入间距的问题。	146	LoadFandol: 当没有设置字体时, 使用 Fandol 字体系列。	133
\xeCJKVerbAddon: 禁止自动换行, 与西文一致。	138	v3.2.11 (2014/03/14 – 2014/04/10)	
v3.2.6 (2013/07/29 – 2013/08/15)		General: 删除 \xeCJKcaption。	161
General: AutoFakeBold 和 AutoFakeSlant 选项直接使用 fontspec 的设置, 修正不能调用相应实际字体的问题。	122	左右角括号 U+2329 和 U+232A 是西文标点符号。	31
case 类函数的用法与 L ^A T _E X3 同步。	1	\CJK@family: 引入 \CJK@family 保存实际的字体族名。	129
为 \mathrm 减少一个可能的数学字体族。	146	indentfirst: 放弃 indentfirst 和 CJKnumber 选项。	142
\AtEndUTFCommand: 可以指定特定符号命令使用的钩子。	197	\xeCJK_add_to_shipout:n: 不再使用内部名字。	24
_xeCJK_boundary_register_text_command:NN: 考虑 ulem 对 \MakeRobust 的不当定义。	146	v3.2.12 (2014/05/12)	
考虑 \math 和 \ensuremath。	146	General: 更新 \int_to_Hex:n。	119
\xeCJK_CJK_and_Boundary:w: 更好的处理边界是 \relax 的情况。	81	新增 RubberPunctSkip 选项。	104
\xeCJK_set_mathfont:: 设置粗体时先检查对应字体是否存在。	135	v3.2.13 (2014/06/02 – 2014/06/20)	
v3.2.7 (2013/08/22 – 2013/11/09)		General: 自动调整 \CJKfamilydefault 时, 只将 \familydefault 展开一次。	133
General: 标点宽度设置禁用比例选项的值改为 nan。	113	\xeCJK_set_mathfont:: 修复参数类型错误。	135
处理 AllowBreakBetweenPuncts 与 xeCJKfntef 的兼容问题。	88	v3.2.14 (2014/10/31 – 2014/11/03)	
实现自定义行首/尾标点符号宽度功能。	103	General: xeCJKfntef 不再依赖 CJKfntef。	162
使用 everypage 往 \shipout 盒子里加钩子。	24	解决下划线前后没有 \CJKglue 或 \JKEcglue 的问题。	162
		完善 \varCJKunderline 的实现。	162
		v3.2.15 (2014/11/07 – 2014/11/10)	
		General: xeCJKfntef 增加 hidden 选项。	162
		把 REVERSE SOLIDUS(U+005C)、HYPHEN-MINUS (U+002D) 和 EN DASH(U+2013) 归入 NormalSpace 类。	31
		增加 HangulJamo 字符类。	30
		\CJKunderanyline: 完善选项。	175

_xeCJK_listings_initial_hook:: 修正 breaklines 无效的问题。	181		
\xeCJKfontof: 完善选项。	174		
v3.2.16 (2014/11/20 – 2014/12/16)			
General: 不再依赖 everypage 宏包。	24		
修复 \hbar。	145		
整理 xCJKecglue 的部分代码。	80		
v3.3.0 (2014/12/26)			
General: 不把 NS 类中的一些有禁则的日文归入 FullRight 类。	31		
不把小写日文假名归入 FullRight 类。	33		
\c_xeCJK_PR_chars_clist: 不把 U+20A9 归入 CJK 的 PR 类。	31		
v3.3.1 (2015/01/22 – 2015/05/08)			
General: IVS 字符类更名为 CM。	29		
删去 fixltx2e 和 amsthm 的冲突补丁。	148		
新选项 WidowPenalty。	98		
\CJKaddEncHook: 应用 0.99992 版的新原语 \Ucharcat。	161		
LoadFandol: 为方便 MacTeX 用户, Fandol 字体改用文件名。	133		
_xeCJK_boundary_register_text_command: NN: 兼容 L ^A T _E X 2 _ε 2015。	146		
\xeCJK_check_single_cs: NNn: 补充可能遗漏的空格。	100		
\c_xeCJK_CM_chars_clist: 补充音调符号。	35		
_xeCJK_listings_initial_hook:: 解决 prebreak 和 postbreak 功能失效的问题。	181		
_xeCJK_listings_process_Default:nN: 对 listings 的字符扩展不影响到其符号表中的七位或八位字符。	183		
\xeCJK_token_value_charcode:N: 0.99992 版修复了 \meaning 的 Bug。	27		
\g_xeCJK_xetex_allocator_int: 兼容 L ^A T _E X 2 _ε 2015。	149		
v3.3.2 (2015/05/15)			
General: 随 Unicode 7.0 更新简繁体映射。	1		
\g_xeCJK_xetex_allocator_int: \xe@alloc@intercharclass 总是有定义的。	149		
v3.3.3 (2015/05/30 – 2016/02/01)			
General: 把 EN DASH (U+2013) 作为半字线连接号归入 FullRight 类。	31		
补充 Ext-E。	33		
不再把 U+2015 和 U+2500 归入 FullRight 类。	31		
更新 L ^A T _E X 3 代码。	1		
兼容 L ^A T _E X 2 _ε 2016/02/01 的字符类设置。	30		
解决与 microtype 宏包的兼容问题。	161		
使用新的 Unicode 编码名称 TU。	188		
CJKfilltwosides: 确保进入水平模式。	180		
v3.3.4 (2016/02/07)			
General: 兼容 X _Y L ^A T _E X 0.99994 的边界字符类。	30		
v3.4.0 (2016/05/01 – 2016/05/13)			
General: RubberPunctSkip 选项有新的值 plus 和 minus。	104		
CJKmath 功能也支持分区字体。	135		
标点符号的压缩量能伸长到原始空白, 能收缩到较小边距。	114		
改进 xCJKecglue 的实现。	44		
\xeCJK_set_mathfont:: CJKmath 的字符范围遵从 \xeCJKDeclareCharClass 的设置。	135		
v3.4.1 (2016/05/21 – 2016/08/18)			
General: 补充 Unicode 9.0 的西夏文。	33		
新的下划线选项 textformat。	162		
修复 CJKspace 功能失效。	81		
v3.4.2 (2016/10/19)			
General: 避免在破折号之间折行。	40		
\xeCJK_clear_Boundary_and_CJK_toks:: 提高效率, 避免重复循环。	39		
v3.4.3 (2016/10/27 – 2016/11/18)			
\CJKfontspec: 允许字体属性可选项在后的新语法。	131		
\setCJKfallbackfamilyfont: 允许字体属性可选项在后的新语法。	120		
\setCJKmathfont: 允许字体属性可选项在后的新语法。	134		
\setCJKmonofont: 允许字体属性可选项在后的新语法。	130		
_xeCJK_long_punct_kerning:N: 考虑破折号边界为负值的情况。	109		
v3.4.4 (2016/11/30)			
General: 不压缩长标点与其他标点的间距。	113		
v3.4.5 (2017/01/02)			
General: 更新 L ^A T _E X 3 的过时用法。	146		
v3.4.6 (2017/02/23)			
\xeCJK@family: 将族名参数完全展开, 以解决与 fontspec 2017/01/24 v2.5d 的兼容问题。	148		
v3.4.7 (2017/03/20)			
General: 简化 CheckSingle 的实现, 不再展开宏。	98		
v3.4.8 (2017/05/15)			
General: 转义 \lstinline 参数中的 _12。	187		
v3.5.0 (2017/07/19 – 2017/07/22)			
General: 补充 Ext-F。	33		
常数 \c_minus_one 已过时。	1		
使用 lazy 函数对 Boolean 表达式进行最小化运算 (L ^A T _E X 3 2017/07/19)。	1		
v3.5.1 (2017/11/16)			
General: 修正 fallback 字体后无法忽略空格的错误。	119		
v3.6.0 (2018/01/13 – 2018/01/24)			
General: Default 类与 MiddlePunct 之间不应该有 \CJKglue。	94		
把 TWO-EM DASH (U+2E3A) 归入 FullRight 类和设为 LongPunct 与 MiddlePunct。	31		
将全角浪线 U+FF5E 等连接号归入 FullRight 类和设为 MiddlePunct。	31		
解决标点中间被隔开的禁则与压缩问题。	92		
同步 L ^A T _E X 3 2017/12/16。	1		
新增 PunctFamily 选项支持对汉字标点单独切换字体。	129		
修正标点同为 LongPunct 与 MiddlePunct 时的实现错误。	94		
总允许长标点与其他标点之间折行。	87		
v3.6.1 (2018/02/25 – 2018/02/27)			
General: 减少 bool 运算。	1		
\xeCJK_if_last_punct:TF: 细化判断。	92		
v3.7.0 (2018/03/12 – 2018/03/18)			
General: 补充定义 \textthyphenationpoint 和 \texttwoemdash。	200		
不再默认引入 xunicode 宏包。	144		
对 \nobreakspace 的恢复放到 xunicode-addon 中处理。	148		

修正长标点被隔开时的压缩处理错误。.....	109	将 CJKfntef 包替换为 xeCJKfntef 包。.....	23
v3.7.1 (2018/04/30)		应用 \disable@package@load 和	
\AtEndUTFCommand: 修复代码重构而引入的新错误。...	197	\declare@file@substitution。.....	23
v3.7.2 (2018/05/02 – 2019/04/07)		v3.8.8 (2021/09/15 – 2021/09/16)	
General: 改用 xparse 的新参数类型 b 定义		General: 补充女书。.....	33
CJKfilltwosides* 环境, 不再依赖 environ 包。.....	180	同步 Unicode 14.0。.....	33
简化 CJKspace 的实现, 并修复错误。.....	81	v3.8.9 (2022/05/26)	
解决与 microtype 宏包的兼容问题。.....	199	General: 修正居中标点悬挂错误。.....	112
删除定义新字体族时过滤重复选项的功能。.....	123	_xeCJK_bound_type_1_glue:Nn: 增加位于段首的支架	
同步 L ^A T _E X3 2019/03/05。.....	1	盒子判断。.....	90
\xeCJK_FullLeft_and_Default:: 再次修正 FullLeft 类		v3.9.0 (2022/06/06 – 2022/07/08)	
字符与西文连用断词失败的问题。.....	89	General: 不直接依赖 xparse 和 l3keys2e。.....	23
_xeCJK_patch_tuenc_composite:: 修复补丁错误。..	151	修复西文的 character protrusion 功能。.....	80
v3.7.3 (2019/04/15)		v3.9.1 (2022/07/28 – 2022/08/02)	
General: 补充日文假名扩展。.....	33	General: 简化部分内部实现。.....	49
修复 penalty 数值错误。.....	92	修复下划线中数学公式的错误处理。.....	167
v3.7.4 (2019/05/31)		v3.10.0 (2022/08/31 – 2026/06/23)	
General: 简化行首/尾标点符号宽度的实现。.....	103	General: NoBreakLongPunct 在右侧时禁止折行。.....	87
v3.8.0 (2020/02/09 – 2020/02/10)		\reset@color 补丁增加 hlist 回退路径, 当 color pop 后	
General: 兼容 L ^A T _E X 2 _ε 2020/02/02 对 NFSS 的修改。.....	1	最后节点为 hlist 时设置	
清理过时的兼容性补丁代码。.....	1	\g__xeCJK_reset_color_pending_bool, 延迟到	
删除 CJKfntef 补丁。.....	161	_xeCJK_check_for_glue_skip: 中处理 \colorbox	
删除 realscripts 补丁。.....	148	等命令右侧间距(#831)。.....	156
删除 \hbar 补丁。.....	145	补丁 l3color 后端的 _color_select:N 和	
删除 \mathrm 补丁。.....	146	_color_backend_reset:, 使 l3color 接口的颜色切换	
应用 \fp_if_nan:nTF。.....	113	也能正确保持 xeCJK 间距(#832)。.....	157
应用 \peek_remove_spaces:n。.....	27	补丁 \reset@color 以在颜色弹出 whatsit 之后重新放	
\@setupverbvisiblespace: 更新可视空格补丁。.....	141	置 xeCJK 节点标记, 修复 \textcolor 命令右侧多余	
v3.8.1 (2020/02/14)		inter-word glue 的问题(#831)。.....	156
General: 修复 \l_xeCJK_current_font_tl 标记错误。..	126	撤回对 \char 的重定义, 改为提供 \xeCJKchar 新命令,	
应用 \shapedefault。.....	136	避免破坏依赖	
v3.8.2 (2020/02/17)		let	
General: 避免导言区字体警告。.....	131	保存 \tn {char} 原语的宏包(\#800)。hdclindex8300153	
修复分区字体错误。.....	127	改正拼写错误 _xeCJK_glue_node:n。.....	49
v3.8.3 (2020/03/15 – 2020/04/27)		将全角浪线等连接号从 MiddlePunct 改为 LongPunct,	
General: hidden 选项保留原内容的高度和深度。.....	166	消除不必要的标点压缩间距。.....	31
补充 U+02EA 和 U+02EB。.....	33	使用 \l_keys_key_str 和 \l_keys_choice_str 替代	
兼容 unicode-math 和 CJKmath 选项。.....	154	已废弃的 _tl 版本(#806)。.....	156
取消 xeCJKfntef 的初始彩色设置。.....	178	提升 L ^A T _E X3 最低版本要求至 2025/10/09。.....	22
删除 _nopar。.....	1	提升版本号至 v3.10.0。.....	1
同步 Unicode 13.0。.....	33	通过在 \Hy@BeginAnnot 中保存并选择性恢复 xeCJK 节	
修复 xCJKecglue 选项。.....	92	点标记, 同时清除旧标记, 解决目录中链接注释起始处的	
依赖 ctexthook 宏包。.....	23	虚假 ecglue(#810), 并为 \ref 提供前侧 ecglue(#809)。155	
重构 PunctStyle 选项, 完全展开参数。.....	116	同步 Unicode 15.0。.....	33
v3.8.4 (2020/05/31 – 2020/06/04)		同步 Unicode 15.1。.....	33
General: 重构后备字体的实现, 修正标点符号无后备字		同步 Unicode 17.0。.....	33
体的问题。.....	118	为 color/xcolor 添加兼容补丁, 在颜色切换 whatsit 后重	
\xeCJK_font_gset_to_current:N: 不缓存 \nullfont。..	25	放 xeCJK 节点标记, 修复 \textcolor 后 CJKecglue 丢	
v3.8.5 (2020/06/25 – 2020/06/26)		失的问题(#315, #803)。.....	156
General: 进一步兼容 microtype。.....	154	为 hypdoc 的 \HD@target 添加补丁, 在它产生的 hbox	
_xeCJK_bound_type_1_glue:Nn: 增加盒子高度判断。..	90	之后重放 xeCJK 节点标记, 修复 l3doc 中 \cs_meta 等	
v3.8.6 (2020/10/17 – 2020/10/18)		命令后 CJKecglue 丢失或保留为原始空格的问题	
General: 兼容 L ^A T _E X 2020/10/01 的 NFSS 钩子机制。...	148	(#873)。.....	157
正确还原标点符号后的 penalty 状态。.....	92	为 mtp ₂ 提供兼容补丁, 使大花括号内部的 \char 不	
v3.8.7 (2021/06/04 – 2021/06/09)		被 interchar 拦截(#407)。.....	153
General: 更好地兼容 CJKnumb。.....	161		

为 url 的 \Url@FormatString 添加补丁, 在进入数学模式前 drain 缓存的 CJKecglue, 修复 CJK 文字与 \url 命令之间间距丢失的问题(#880)。	157	glue 检查, 修复前 {中} 后类型排版中多余的 \CJKecglue(#831)。	81
文档字体从 Source Han Serif 统一为 Noto Serif CJK (#686)。	1	修复 CJK 字符后行尾空格在宏命令前未被消除的问题, 不再立即输出空格 glue, 改为只留标记由后续恢复路径决定间距(#324)。	81
新增 NoBreakLongPunct 属性, 禁止在省略号等长标点前断行。	94	\xeCJK_fntef_sbox:n: 保存并恢复	
新增 \xeCJKchar, 绕过 interchar 机制(#407)。	153	\g__xeCJK_last_node_tl, 避免 hbox 内渲染装饰符号时 interchar toks 污染全局节点标记, 修复 xeCJKfntef 与 \textcolor 组合使用时多余的 CJKecglue。	177
新增实验性 experiment/halfright-prebreakpenalty 选项(#811)。	84	\xeCJK_make_node:n: 新增	
修复 xCJKecglue 错误。	49	\g__xeCJK_glue_check_pending_bool 用于标记 xeCJKfntef 组结束后的首次 glue 探测。	50
修复 \set@color 补丁在无真实节点标记时未清除全局状态, 导致首次 \textcolor 在标点或段首后插入虚假 ecglue 的问题(#807)。	156	新增 \g__xeCJK_reset_color_pending_bool 用于标记 \reset@color 在 hlist 上下文中触发, 延迟到	
修复 \textcolor 包裹 ulem 类下划线命令时 CJK 字间距异常的问题(#830)。	1	__xeCJK_check_for_glue_skip: 中处理 \colorbox 等命令右侧的 color pop whatsit(#831)。	50
修复 \textcolor 等 whatsit 节点导致 CJKecglue 丢失的问题。	49	\c__xeCJK_middle_dot_prop: 在 \Pifont 中先进入水平模式, 防止 \makeCJKinactive 在垂直模式下通过分页泄漏到输出例程(#688)。	153
修复 \textnleqslant 的定义, 修复 \textnbacksim 和 \textnlessapprox 定义里的自引。	203	__xeCJK_punct_boundary_guard:: 新增	
修复 \textsbleftarrow 的定义。	200	experiment/punct-measure-fix 选项: 段落模式下通过 para/end 钩子恢复被 \unskip 移除的标点补偿 glue (#859)。	85
修复 \xeCJKnobreak 错误。	49	修复 tabular 中 \unskip 吞掉标点补偿 glue 的问题 (#827)。	85
修复拼错的命令名, \cyrreref → \cyrrerev 和 \textDiamandSolid → \textDiamondSolid。	203	__xeCJK_skip_if_interword:N: 移除通用 whatsit 恢复, 避免 hyperref 链接注释等 whatsit 节点导致 CJKecglue 在错误位置插入(#803)。配合 \set@color 定点补丁保持 \textcolor 场景的正确性(#315)。	42
修复颜色补丁在 \g__xeCJK_last_node_tl 被无关的 \set@color 清空后仍尝试重放节点导致 Missing number 错误的问题, 影响 listings 与 \rulecolor 组合 (#836)。	156	\xeCJK_kunderdot: 移除 \ignorespaces, 修复命令后空格丢失 (#465)。	178
修复在 \lstinline 参数中使用非 # 的 catcode~6 字符(如 \texttt {\textbackslash catcode~6}) 时输出错误的 TeX 模式请求设置		\g__xeCJK_glue_check_pending_bool, 修复 \xeCJK_kunderdot 等命令右侧空格问题 (#826)。	179
hdclin-dex10081181		\xeCJKsetcharclass: Deprecated:	
修复字体切换组(如 \texttt、\zihao)导致 CJKecglue 使用错误字体度量的问题。	49	\xeCJKsetcharclass 已弃用, 调用时报错并提示改用	
修复组合符号 U+04AA 的定义。	203	\xeCJKDeclareCharClass(#709)。	145
在下划线内 CJK 分组切换时保存/恢复字体状态 (#465)。	168	v3.10.1	
\CJKfontspec: Breaking: \newCJKfontfamily 定义的字体切换命令改为局部定义, 与 \newcommand 行为一致。在分组内调用时, 命令不再泄漏到分组外 (#751)。	131	(2026/06/29 – 2026/07/01)	
\CJKkunderansymbol: 移除 \ignorespaces, 修复命令后空格丢失 (#465)。	178	General: 提升版本号至 v3.10.1。	1
\CJKkunderdot: 移除 \ignorespaces, 修复命令后空格丢失 (#465)。	178	为 l3doc 的 __codedoc_meta:n 添加补丁, 将参数内容包入 \hbox:n, 修复 \meta 内首字符前出现多余 CJKecglue 的问题 (#920)。	159
\xeCJK@family: Breaking: 移除对 L ^A T _E X 2 _ε 2020/10/01 之前版本的字体钩子兼容代码 (#746)。	148	为 L ^A T _E X 核心命令 \verb 添加补丁, 在进入 verb 内容前 drain 缓存的 CJKecglue, 修复 CJK 文字与 \verb(或 shortvrb 的短记号)之间间距丢失的问题 (#910)。	158
\xeCJK_check_for_glue:: 当 glue 下方是 hlist(\mbox 等命令产生的 hbox)时, 不依赖 boolean, 直接通过 \g__xeCJK_last_node_tl 判断前方内容类型, 修复 \mbox、\colorbox 等 hbox 命令右侧多余 inter-word glue (#831)。	46	为 biblatex 的 \blx@pagetracker 添加补丁, 清空 \g__xeCJK_last_node_tl, 修复中文参考文献条目首字符前出现多余 CJKecglue 的问题 (#931)。	160
新增 glue 分支处理, 修复 xeCJKfntef 命令右侧空格问题。	46	v3.10.2	
\xeCJK_CJK_and_Boundary:w: 当显式} 触发 CJK→Boundary 转换时, 设置 \g__xeCJK_glue_check_pending_bool 以启用后续的		(2026/07/03 – 2026/07/06)	
		General: PoZheHao 类字符同样按全角右标点处理 (#382)。	105
		\verb 组关闭时用 \setlanguage 主动排出 language whatsit, 修复 \verb 右侧源码空格场景下 CJKecglue 双倍的问题 (#919)。	158
		提升版本号至 v3.10.2。	1
		为 doc 的 \meta 添加补丁, 将参数内容包入 \hbox:n, 修复 \meta 内首字符前出现多余 CJKecglue 的问题 (#951)。	159

文档等宽字体改用朱雀仿宋, 意大利体改用霞鹜文楷 GB Lite(#908)。	1	把 \HD@target 迁移到统一 transparent capture, 同时恢复其前后的 marker 与源码空格状态(#992)。	157
文档字体启用 Language 与 PoZheHaoLigature, 破折号按全角字形合字输出(#382)。	1	把 \verb 从假定西文输出的入口 drain 迁移到统一 stream capture, 按 verbatim 实际首尾类别恢复两侧边界(#992)。	158
新增 LatinPunct 选项, 中西文共用的弯引号、间隔号与省略号可以切换为西文标点(#389、#431)。	104	把链接 annotation 的入口保存和末尾定点重放迁移到统一 stream capture, 按链接实际首尾可见类别恢复两侧边界(#992)。	155
新增 PoZheHaoLigature 选项, 支持字体的破折号合字功能(#382)。	104	补充 Boundary→CJK 方向的说明: 某些显式 glue 也可能被替换为 (\#996)。hdclindex163	
增加 PoZheHao 字符类, 支持字体的破折号合字功能(#382)。	30	将 I3color 的颜色 push/pop 入口注册为 transparent capture, 删除对应专用 marker 保护分支(#992)。	157
长标点与其他标点之间折行时检查断点两侧的禁则: 左标点不得悬于行尾, 右标点与 NoBreakLongPunct 不得落于行首(#456)。	87	将 \set@color 与 \reset@color 注册为 transparent capture, 删除颜色 push/pop 的专用 marker save/replay 分支(#992)。	156
_xeCJK_long_punct_kerning:N: 破折号中间的压缩量改为多路取大, 保证两个 U+2014 的总宽不超过两个汉字宽(#382)。	109	明确 xCJKecglue 同时控制行内数学公式旁的源码空格(#1002)。	3
_xeCJK_punct_if_full_margin_dash:N: 新增判断, 配合破折号占两个汉字宽的空白补偿(#382)。	110	提升版本号至 v3.10.4。	1
\xeCJK_punct_margin_process:NN: 未启用合字的破折号两端各补偿一整份空白, 保证连用的破折号占两个汉字宽(#382)。	111	新增 siunitx 兼容: \unit、\qty、\num 与 v2 旧名 \si、\SI 注册为固定 Default 首尾的 stream capture, 修复 CJK 上下文中单位命令边界的空格丢失(#1000)。	157
\xeCJKResetPunctClass: 重置后恢复 LatinPunct 的设置(#389)。	37	修正 _xeCJK_check_num_range:nnNN 中缺省端点的范围解析问题。	36
重置后恢复 PoZheHaoLigature 的设置(#382)。	37	用固定 Default 首尾的 stream capture 包围 \Url@z 的完整格式化阶段, 删除 #880 的专用 drain(#992)。	157
v3.10.3 (2026/07/12 – 2026/07/13)		在用户手册说明 $\TeX$ 无法区分源码空格与某些显式 glue, 并给出前置零宽 \kern 的处理方法(#992)。	3
General: 将朝鲜文字母分为 L/V/T 三类, 在音节内保持 shaping, 在相邻音节间恢复 \CJKglue(#158)。	41	_xeCJK_boundary_replay_before:: box/wrapped-box 捕获在盒子直接排出了可见内容(宽度非零、高度或深度非零, 且末节点为 char/rule/math/kern)而未观察到任何字符类别时, 按 Default 首尾重建边界; math 与 \vrule 等不触发 interchar 转换的可见内容不再被误判为“无可见输出”。	
开明式句末点号的宽度由 0.8em 改为全角(#975)。	117	已经排好再放入的盒子与只用于留白的盒子仍按无可见输出处理(#998)。	50
全角格式启用相邻标点边界优化, 并保留左标点后接右标点的自然空白(#975)。	117	新增独立的行内公式边界类别, 分别处理公式末尾与公式后已有参数内源码空格, 使注册命令能按直接公式的源码空格语义恢复左右边界; 另行区分仍参与外层断行的 stream 空格与已冻结在盒子中的空格, 正确保留弹性间距的伸缩量; 不可见节点将普通 stream 的真实空格与 marker 隔开时按直接公式语义让 marker 过期, 并保留中间 glue 与节点的原有次序(#1002)。	50
提升版本号至 v3.10.3。	1	新增命令边界 capture/register 框架, 按实际可见输出的首尾类别统一处理盒子、不透明节点流和无可见输出命令(#992)。	50
新增 CJLineBreak 选项, 为日文小假名提供可选严格行首禁则(#165)。	104	新增实验性 experiment/boundary-register 接口, 允许专家用户把自定义命令注册到统一命令边界框架(#1010)。	50
新增 CJStarter 类和可选严格行首禁则, 保持小假名的普通 CJK 字距(#165)。	41	重放 Default marker 时按外层列表同步 \spacefactor; post-transparent 同时搬移 marker 与其后的候选 glue, 修复 xCJKecglue 启用后盒内末尾大写字母和 \null 遮住边界后缀时的右边界间距(#1003)。	50
新增 enabled-left-right-kerning 标点格式选项(#975)。	15	\xeCJK_check_for_glue:: Boundary 到 CJK 的源码空格检查改用与 Default 方向相同的	
在顶层 \Hy@EndAnnot 结束前检测 URL 的数学节点, 并在链接结束 whatsit 后放置可信的西文边界标记, 修复 \url 右侧的 CJKecglue 丢失问题(#972)。	155	_xeCJK_skip_if_interword:N 判断函数, 自然宽度	
\CJKsout: 使用 \cleaders 使删除线相对正文居中(#967)。	175		
\CJKunderanyline: 使用 \cleaders 使自定义线相对正文居中(#967)。	175		
\CJKunderdblline: 使用 \cleaders 使双下划线相对正文居中(#967)。	174		
\CJKunderline: 使用 \cleaders 使下划线相对正文居中, 避免起始位置改变时产生水平偏移(#531)。	174		
\CJKunderwave: 使用 \xleaders 使波浪线相对正文对齐, 同时保持字间连接连续(#967)。	174		
\CJKxout: 使用 \cleaders 使斜删除线相对正文居中(#967)。	175		
v3.10.4 (2026/07/18 – 2026/07/22)			
General: 把 \blx@pagetracker 从单向清空状态改为 transparent capture, 使 write whatsit 对两侧实际可见边界透明(#992)。	160		

不等于当前词间空格的显式 glue 不再被替换为 \CJKglue(#996)。.....	46
由命令边界 framework 接管 xeCJKfntef、盒子命令和颜色命令,删除按全局 tl 猜测 hlist/whatsit 下方类别的旧回退;glue 分支现在只接受紧邻的 xeCJK marker(#992)。	46
\xeCJK_fntef_sbox:n: 改用 framework 的可嵌套暂停区隔离测量装饰符号时构造的盒子,同时保存、恢复 marker 与 pending 状态(#992)。	177
_xeCJK_listings_boundary_stream_end:: 将 \lstinline 的分隔符、活动字符和花括号扫描路径接入 auto stream capture,使颜色切换前后的四种源码空格组合都与直接输入一致(#992)。	187
\xeCJK_make_node:n: 删除颜色专用 pending boolean;颜色 push/pop 和盒子输出改由 transparent/wrapped-box capture 处理(#992)。	50
_xeCJK_skip_if_interword:N: 新增	
_xeCJK_glue_check_expire_stale:: 顶层恢复链发现节点列表为空时清除过期的 pending 状态,阻止 \g_xeCJK_glue_check_pending_bool 跨 \hbox/\setbox 存活(#996)。	42
允许 Boundary 到 Default 的恢复路径暂时移除末尾不含无限阶伸缩、且有收缩量的源码词间 glue;若其下方确有 CJK marker,则将其替换为 \CJKecglue,补齐与 Default 到 CJK 方向对称的命令边界处理(#992)。	42
\xeCJK_ulem_on:n: 将 xeCJKfntef 线型命令、原生 ulem 入口(如 \uline)、\xeCJKfntefon 与独立符号命令接入 stream capture,并让内部 pending 通过 framework 的统一辅助函数发布;删除 fntef 专用末状态恢复和 pending 设置(#992)。	172
让嵌套的 ulem/xCJKfntef 线型命令只由最外层启动 stream capture,修复 capture 栈逐次累积的状态泄漏 (#992)。	172

代码索引

意大利体的数字表示描述对应索引项的页码;带下划线的数字表示定义对应索引项的代码行号;罗马字体的数字表示使用对应索引项的代码行号。

Symbols

\#	3156, 3157
\(	1448
\)	1487
\\	5, 6, 7, 13, 14, 15, 16, 31, 366, 367
\(font-switch)	10

A

\addCJKfontfeatures	10
\AddToHook	2784, 2786, 2797, 2799, 2810, 2812, 2823, 2833, 2835, 3199, 3201, 3532, 3534
AllowBreakBetweenPuncts	6
\AtBeginDocument	70
\AtEndOfPackage	433, 729, 742, 753, 777, 807
AutoFakeBold	4, 9
AutoFakeSlant	4, 9
AutoFallBack	4

B

bool commands:

\bool_gset_false:N	929, 945, 957, 1093, 1103, 1132, 1149, 1714, 1829, 2068, 2156, 2298, 2401, 2531, 3530, 3533, 3539
\bool_gset_true:N	1713, 2776, 3041, 3317, 3517
\bool_if:NTF	102, 320, 610, 612, 633, 902, 944, 1080, 1090, 1094, 1101, 1109, 1119, 1131, 1135, 1148, 1180, 1700, 2003, 2006, 2268, 2886, 2905, 2907, 2989, 2999, 3293, 3337, 3347, 3355, 3379, 3404, 3408, 3462, 3469, 3515, 3536
\bool_lazy_and:nnTF	2315
\bool_lazy_or:nnTF	557
\bool_lazy_or_p:nn	2321

\bool_new:N	53, 111, 270, 615, 616, 637, 727, 1309, 1317, 1318, 1319, 3265, 3285, 3459, 3508
\bool_set_false:N	245, 2889, 2890, 3239, 3277
\bool_set_true:N	104, 252, 723, 2898, 2903, 2921, 2927, 2945, 2951, 2961, 2974, 2983, 2993, 3003, 3228, 3250, 3271
Boundary	369
box commands:	
\box_clear:N	2346, 2383
\box_dp:N	1886, 2328
\box_ht:N	1885, 2324
\box_if_exist:N	1336
\box_new:N	51, 1325, 1326, 1327, 1328, 1337, 2305
\box_set_to_last:N	1881, 2292
\box_use_drop:N	1890, 1918, 1922, 1940, 1948, 1952, 2447, 2464
\box_wd:N	224, 1884, 2318
boxdepth	18

C

\changes	3155
char commands:	
\char_generate:nn	436
\char_set_catcode_ignore:n	332, 333, 334, 335, 336
\char_set_catcode_letter:n	575
CheckFullRight	6
CheckSingle	4
CJK	369
\CJKecglue	841, 1004, 1398, 1402, 1983, 1990, 3339, 3397, 3405
CJKecglue	3, 3218
\CJKfamily	10
\CJKfamilydefault	11
CJKfilltwosides	19

`\CJKfontspec` 10
`\CJKglue` 805, 1111,
 1121, 1157, 1161, 1183, 1226, 1983, 1994, 3213, 3279, 3433
`CJKglue` 3, 3209
`CJKmath` 3
`\CJKrmdefault` 10
`\CJKsfdefault` 10
`\CJKsout` 17
`CJKspace` 3, 3266
`\CJKsymbol` 796, 828, 1045, 3435, 3441, 3447
`\CJKttdefault` 10
`\CJKunderanyline` 18
`\CJKunderanysymbol` 19
`\CJKunderdblline` 17
`\CJKunderdot` 17
`\CJKunderline` 17
`\CJKunderwave` 17
`\CJKxout` 17
`CJLineBreak` 6
`CJStarter` 388
clist commands:
`\clist_const:Nn` 400, 402, 404, 405, 412, 414, 419, 427,
 429, 431, 441, 443, 444, 453, 454, 463, 513, 520, 522, 524
`\clist_gclear:N`
 602, 603, 604, 605, 619, 620, 621, 622, 623, 624, 625, 626
`\clist_gconcat:NNN` 540, 554
`\clist_if_in:nnTF`
 ... 2480, 2542, 2577, 2610, 2669, 2672, 2731, 2757, 2770
`\clist_map_function:NN` 438
`\clist_map_function:nN` 714
`\clist_map_inline:Nn` 542, 592
`\clist_map_inline:nn`
 ... 34, 39, 755, 766, 768, 818, 836, 988, 3164, 3451, 3522
`\clist_new:N` 55, 346, 358
`\clist_set:Nn` 535
CM 388
`\cs` 1473, 1476, 2840
cs commands:
`\cs_end:` 308
`\cs_generate_variant:Nn` 554, 644, 657, 663, 2629
`\cs_gset_eq:NN` 131, 142
`\cs_gset_protected:Npe` 692
`\cs_gset_protected:Npn` .. 3115, 3131, 3146, 3169, 3181
`\cs_if_eq:NNTF` 142
`\cs_if_exist:NTF` .. 45, 95, 3091, 3127, 3142, 3166, 3178
`\cs_if_exist_use:NTF` 1254
`\cs_new:Npn` 112, 117, 201,
 271, 273, 280, 286, 297, 311, 362, 526, 645, 1281, 1283, 3364
`\cs_new_eq:NN` 267, 268, 269,
 363, 728, 951, 960, 1021, 1185, 3113, 3168, 3180, 3203, 3263
`\cs_new_protected:Npe` 3204
`\cs_new_protected:Npn`
 ... 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67,
 73, 75, 77, 79, 81, 83, 85, 92, 108, 128, 130, 132, 139,
 141, 210, 213, 229, 234, 238, 249, 319, 339, 352, 533,
 538, 555, 579, 589, 639, 650, 652, 658, 664, 677, 688,

690, 706, 720, 799, 843, 848, 854, 860, 900, 914, 940,
 947, 952, 961, 1008, 1010, 1016, 1047, 1064, 1076, 1086,
 1129, 1141, 1146, 1152, 1172, 1178, 1186, 1201, 1225,
 1259, 1266, 1276, 1285, 1290, 1334, 1360, 1389, 1409,
 1522, 1528, 1560, 1571, 1588, 1615, 1646, 1674, 1684,
 1703, 1718, 1734, 1743, 1762, 1771, 1774, 1798, 1843,
 1861, 1873, 1911, 1932, 1960, 2071, 2081, 2087, 2105,
 2148, 2162, 2177, 2187, 2218, 2262, 2282, 2288, 2366,
 2388, 2456, 2470, 2498, 2528, 2534, 2540, 2546, 2575,
 2631, 2637, 2643, 2651, 2657, 2667, 2684, 2725, 2768,
 2779, 2791, 2804, 2817, 2828, 2844, 2846, 2884, 2912,
 2931, 2949, 2965, 3010, 3039, 3056, 3087, 3104, 3110,
 3125, 3140, 3162, 3287, 3335, 3341, 3343, 3365, 3371,
 3385, 3387, 3393, 3400, 3424, 3431, 3481, 3499, 3510, 3542
`\cs_set:Npn` 276, 435, 1228, 1234
`\cs_set_eq:NN` 129,
 134, 135, 136, 240, 3130, 3145, 3229, 3230, 3231, 3232,
 3233, 3240, 3243, 3244, 3253, 3254, 3255, 3256, 3257, 3391
`\cs_set_protected:Npe` 241, 243
`\cs_set_protected:Npn` ... 3213, 3222, 3251, 3272, 3278
`\cs_to_str:N` 2848, 2917, 3108, 3114, 3118
`\cs_undefine:N` 137, 439
ctex commands:
`\ctex_after_end_preamble:n` 72
`\ctex_at_end_package:nn` 86
`\ctex_at_end_preamble:n` 71
`\ctex_disable_package:n` 22, 41, 44
`\ctex_if_format_at_least:nTF` 42
`\ctex_patch_cmd:Nnn` 22
`\ctex_replace_package:nn` 43

D

Default 369
`\defaultCJKfontfeatures` 10
depth 18
dim commands:
`\dim_case:nn` 1154, 3410
`\dim_case:nnTF` 1104, 2110
`\dim_compare:nNnTF` 888, 891, 1088, 1882
`\dim_compare_p:nNn` 2317, 2323, 2327
`\dim_const:Nn` 1263
`\dim_eval:n` 203
`\dim_gset:Nn` 1263, 3518
`\dim_if_exist:NTF` 1262
`\dim_max:nn` 2228
`\dim_new:N` 52, 880, 1421, 1422, 1423, 1424, 3509
`\dim_set:Nn` 885, 887, 2165,
 2167, 2179, 2181, 2222, 2224, 2226, 2243, 2245, 2247, 3546
`\dim_set_eq:NN` 2193, 2197
`\dim_use:N` 224
`\dim_zero:N` 2189, 2190
`\c_zero_dim`
 ... 892, 1888, 1985, 2171, 2205, 2233, 2318, 2324, 2328

E

else commands:
`\else:` 146, 256, 261, 309, 317, 1240, 1257

EmboldenFactor	4
\ensuremath	1452, 1495
EnvCS	4
EnvCS+	4
EnvCS-	4
exp commands:	
\exp:w	3369
\exp_after:wN	140, 235, 253, 254, 259, 260, 262, 263, 274, 302, 3367, 3368, 3487, 3494, 3503
\exp_args:Nc	344, 1278, 1695, 2064, 2077, 2292, 3113
\exp_args:Ncc	1230
\exp_args:NcNc	3106
\exp_args:Ne	1437, 1498, 1966, 2514, 2555, 2709, 2848
\exp_args:NNNo	219, 223, 226
\exp_args:NNV	3019
\exp_args:NnV	3067, 3072, 3077
\exp_args:NNVV	3031
\exp_args:NV	1434, 1443, 1447, 1451, 1467, 1494, 1821, 1837, 1895, 1926, 1953, 2274, 2384, 2451, 2488, 3012
\exp_args:NVV	2444
\exp_end_continue_f:w	3369
\exp_last_unbraced:Ne	114
\exp_not:N	81, 242, 244, 258, 697, 3454, 3456
\exp_not:n	242, 244, 416, 417, 456, 457, 458, 459, 460, 461, 655, 661, 675, 685, 694
\exp_stop_f:	145, 316
experiment/boundary-register	8
experiment/halfright-prebreakpenalty	7
experiment/punct-measure-fix	7

F

FallBack	11
\fbox	3193
fi commands:	
\fi:	146, 235, 264, 265, 309, 317, 1240, 1257
format	18
fp commands:	
\fp_compare:nNnTF	372
\fp_eval:n	204
FullLeft	369
FullRight	369

G

gap	18
group commands:	
\group_align_safe_begin:	246, 3292, 3417
\group_align_safe_end:	242, 244, 3342, 3395, 3420, 3426
\group_begin:	215, 275, 1227, 3106
\c_group_begin_token	722, 2084, 3360
\group_end:	226, 303, 1250, 3106
\c_group_end_token	728, 2284, 3360

H

HalfLeft	388
HalfRight	388
HangulJamo	388
HangulJamoL	388

HangulJamoT	388
HangulJamoV	388

hbox commands:

\hbox_set:Nn	2340, 2374
\hbox_set:Nw	216
\hbox_set_end:	219, 223
\hbox_unpack:N	2342, 2376
\hbox_unpack_drop:N	2460
height	18
hidden	18

I

if commands:

\if_catcode:w	257
\if_cs_exist:w	306
\if_dim:w	1253
\if_int_compare:w	232, 315, 1239
\if_meaning:w	251
\IfBooleanT	531
InlineEnv	4
InlineEnv+	4
InlineEnv-	4

int commands:

\int_case:nnTF	2352
\int_compare:nNnTF	150, 163, 183, 375, 954, 956, 1362, 1365, 1617, 1620, 1648, 1651, 1720, 1745, 1776, 1845, 1963
\int_const:Nn	69, 357, 380, 386, 1236
\int_div_truncate:nn	206
\int_eval:n	281, 288, 298, 312
\int_gdecr:N	1715, 1839, 1869, 2300, 2453
\int_gincr:N	1261, 1268, 1686, 1962
\int_gset:Nn	211, 1274, 2343
\int_gset_eq:NN	992, 1025, 2737
\int_if_exist:NTF	341, 354
\int_incr:N	576, 586
\int_max:nn	566
\int_min:nn	565
\int_new:N	50, 209, 236, 237, 1273, 1310, 1311, 2304
\int_set:Nn	561, 565, 566, 571, 572, 582, 591
\int_set_eq:NN	562
\int_step_inline:nn	1368, 1654
\int_until_do:nNnn	26
\int_use:N	371, 1594, 1601, 1608, 1626, 1632, 1638, 1688, 1691, 1693, 1696, 1699, 1708, 1711, 1756, 1787, 1810, 1815, 1854, 1967, 1970, 1973, 1977, 1979, 1981, 1988, 1992, 1996, 1999, 2002, 2005, 2008, 2012, 2018, 2031, 2040, 2049, 2055, 2060, 2065, 2075, 2078, 2095, 2098, 2293, 2313, 2371, 2393, 2398, 2442, 2448, 2461, 2465, 2477, 2509, 2519, 2537, 2654, 2689, 2696, 2704, 2714
\int_value:w	207
\int_zero:N	725
\c_one_int	329, 375
\c_zero_int	66, 331, 954, 956, 1363, 1366, 1618, 1621, 1649, 1652, 1721, 1746, 1777, 1846, 3513

K

KaiMingPunct	5
--------------------	---

KaiMingPunct+ 5
 KaiMingPunct- 5
 keys commands:
 \keys_define:nn 321, 3209, 3218, 3266
 \keys_set:nn 611, 613, 2894

L

\lastnodetype 2347
 \LaTeXe 3158
 LatinPunct 6
 LoadFandol 9
 LocalConfig 3
 LongPunct 5
 LongPunct+ 5
 LongPunct- 5

M

\makexeCJKactive 324, 328
 \makexeCJKinactive 325, 328
 Mapping 9
 \mbox 3192
 MiddlePunct 5
 MiddlePunct+ 5
 MiddlePunct- 5

mode commands:

 \mode_if_horizontal:TF 1391, 1849, 1875
 \mode_if_inner:TF 3512
 \mode_if_math:TF 1724, 1749, 1780
 \mode_leave_vertical: 1727, 1752, 1783

msg commands:

 \msg_critical:nn 10
 \msg_critical:nnn 20
 \msg_error:nn 58
 \msg_error:nnn 37, 59
 \msg_info:nnnn 64
 \msg_new:nnn 3, 11, 29, 56
 \msg_new:nnnn 57
 \msg_warning:nn 60
 \msg_warning:nnn 61
 \msg_warning:nnnn 62
 \msg_warning:nnnnn 63

N

\newCJKfontfamily 10
 \NewDocumentCommand 45, 100, 328, 330, 528, 552, 595, 600, 617
 NewLineCS 4
 NewLineCS+ 4
 NewLineCS- 4
 \newXeTeXintercharclass 344
 NoBreakCS 6
 NoBreakCS+ 6
 NoBreakCS- 6
 NoBreakLongPunct 5
 NoBreakLongPunct+ 5
 NoBreakLongPunct- 5
 NormalSpace 388
 \normalspacedchars 14, 595

P

peek commands:

 \peek_after:Nw 247, 253, 3367
 \peek_meaning_remove:NTF 993, 1026
 \l_peek_token 251, 258, 999, 1032, 3295,
 3298, 3303, 3309, 3312, 3315, 3321, 3324, 3373, 3376, 3418

PlainEquation 4

PoZheHao 388

PoZheHaoLigature 5

prg commands:

 \prg_do_nothing:
 129, 131, 268, 269, 651, 697, 951, 1185, 3203
 \prg_new_conditional:Npnn
 23, 143, 304, 313, 881, 1237, 1251, 1549, 2137, 2850
 \prg_new_protected_conditional:Npnn
 1426, 1461, 1478, 1510, 1538, 2307, 2334
 \prg_return_false: 26, 146, 309, 317,
 894, 896, 898, 1240, 1257, 1432, 1440, 1454, 1470, 1482,
 1501, 1504, 1518, 1545, 1556, 2144, 2332, 2360, 2362, 2857
 \prg_return_true:
 26, 146, 309, 317, 893, 1240, 1257, 1439,
 1445, 1449, 1453, 1469, 1485, 1488, 1496, 1500, 1514,
 1517, 1546, 1552, 1555, 2140, 2143, 2331, 2359, 2853, 2856

prop commands:

 \prop_gput:Nnn . 2783, 2795, 2808, 2821, 2832, 2845, 3031
 \prop_if_in:NnTF
 2781, 2793, 2806, 2819, 2830, 2852, 2855, 3019
 \prop_map_inline:Nn 3042
 \prop_new:N 1313, 1314, 1315

PunctBoundWidth 6

PunctFamily 5

PunctStyle 5

PunctWidth 6

Q

quark commands:

 \q_stop 274, 276, 280, 286, 297, 302

R

\RequirePackage 21, 22, 46, 47

reverse commands:

 \reverse_if:N 232

RubberPunctSkip 6

S

scan commands:

 \scan_stop:
 25, 28, 216, 240, 320, 1393, 1805, 2022, 3309, 3390, 3391
 \s_stop 115, 117

sep 18

seq commands:

 \seq_gpop:NN 1736, 1764, 1800, 1863
 \seq_gpush:Nn 1722, 1725,
 1728, 1747, 1750, 1753, 1778, 1781, 1784, 1847, 1851, 1857
 \seq_gput_right:Nn 347, 348, 359, 708, 3051
 \seq_gset_eq:NN 701
 \seq_gset_from_clist:Nn 703

`\seq_map_inline:Nn` . . . 731, 744, 757, 779, 809, 3089, 3449
`\seq_new:N` 337, 338, 700, 702, 705, 1312, 1316, 1324
`\seq_pop_left:NN` 3059, 3061
`\seq_set_split:Nnn` 3058
`\setCJKfallbackfamilyfont` 11, 12
`\setCJKfamilyfont` 9
`\setCJKmainfont` 9
`\setCJKmathfont` 11
`\setCJKmonofont` 9
`\setCJKsansfont` 9
`skip` 17
skip commands:
`\skip_const:Nn` 1270
`\skip_gset:Nn` 1270
`\skip_horizontal:N` 865,
868, 943, 949, 966, 973, 974, 981, 983, 1056, 1095, 1096,
1110, 1120, 1136, 1137, 1143, 1163, 1165, 1182, 1196,
1211, 1214, 1217, 1218, 1223, 1583, 1939, 1947, 1955, 2278
`\skip_horizontal:n` 2648
`\skip_if_eq:nnTF`
. 152, 161, 169, 186, 970, 1203, 2276, 2503, 2566
`\skip_if_exist:NTF` 1269
`\skip_if_finite:nTF` 883
`\skip_new:N` 54, 987, 1417, 1418, 1419, 1420, 3217, 3264
`\skip_set:Nn` 220, 224,
227, 886, 2164, 2169, 2203, 2235, 2252, 2270, 2561, 2563
`\skip_set_eq:NN`
. 904, 963, 1066, 1190, 1575, 1934, 2091, 2272, 2502
`\skip_use:N`
. 220, 227, 1611, 1989, 1993, 1997, 2034, 2043, 2096
`\c_zero_skip` 152, 161, 169, 186, 2277, 2567
`SlantFactor` 4
`\SplitArgument` 553
str commands:
`\c_backslash_str` 3016, 3023, 3047, 3094
`\str_case:nn` 1395, 1789, 2967, 3063
`\str_case:nnTF` 2933, 2952
`\str_const:Nn` 370
`\str_if_eq:nnTF` 17, 544, 733, 737,
746, 759, 781, 811, 1826, 2220, 2413, 2458, 2548, 2553,
2579, 2592, 2594, 2612, 2675, 2677, 2679, 2738, 2745, 2751
`subtract` 18
`symbol` 18
sys commands:
`\sys_if_engine_xetex:TF` 10

T

$\TeX$ and $\LaTeX_{2\epsilon}$ commands:

`\(` 21, 50
`\)` 21
`\@beginDvi` 24, 91, 95, 98
`\@empty` 24
`\@ifpackagelater` 19
`\@ifframebox` 3180, 3181
`\@imakebox` 3168, 3169
`\@pkgextension` 28

`\[` 4
`\addCJKfontfeatures` 10
`\AtBeginDocument` 8, 51
`\AtBeginDvi` 24
`\AtBeginShipout` 20, 24
`\baselineskip` 3
`\begin` 4, 19
`\begingroup` 38
`\box` 51
`\catcode` 20, 27
`\char` 21
`\CJKecglue` 8, 42, 44, 51
`\CJKfamily` 10, 20
`\CJKfamilydefault` 11, 12, 14
`\CJKfontspec` 10
`\CJKglue` 8, 20, 41, 51
`\CJKrmdefault` 11
`\CJKsout` 18, 19, 21
`\CJKsymbol` 20
`\CJKunderanyline` 18
`\CJKunderanysymbol` 18, 19
`\CJKunderdblline` 18, 21
`\CJKunderdot` 17–19, 21
`\CJKunderline` 17, 18, 21
`\CJKunderwave` 18, 21
`\CJKxout` 21
`\color` 18
`\color@b@x` 51
`\colorbox` 21
`\cprotect` 20, 21
`\defaultCJKfontfeatures` 10
`\document` 24
`\dotfill` 44
`\emph` 21
`\end` 4, 19
`\endgroup` 38
`\ensuremath` 21, 50
`\ExplSyntaxOff` 8
`\ExplSyntaxOn` 8
`\familydefault` 11
`\fbox` 21
`\fontdimen7` 42
`\footnote` 6, 20
`\footnotemark` 6
`\framebox` 21
`\hbox` 19, 85
`\hrulefill` 44
`\hskip` 3, 42
`\hyperref` 21, 22
`\icprotect` 20
`\kern` 7, 42, 46
`\lastskip` 42
`\leaders` 44
`\makeatletter` 8
`\makeatother` 8
`\makebox` 21, 51

\maxdimen	15, 16	\xeCJKOffVerbAddon	20
\mbox	3, 8, 21, 51	\xeCJKResetCharClass	6
\meaning	27	\xeCJKRestoreSubCJKBlock	14
\newCJKfontfamily	5, 10	\xeCJKsetkern	15–17
\nfss@text	2424	\xeCJKsetup	2, 3, 15, 17–19
\nobreak	6	\xeCJKsetwidth	6, 14, 15
\normalfont	11	\xeCJKShipoutHook	20
\normalspacedchars	14	\xeCJKVerbAddon	9, 20
\null	50, 51	\XeTeXdashbreakstate	40
\outer	20, 21	\XeTeXinterchartoks	21, 26
\par	4, 7	tex commands:	
\parfillskip	7	\tex_font:D	140, 145, 154, 155, 156, 166, 173, 174, 179, 180, 191
\path	3	\tex_fontdimen:D	154, 155, 156, 166, 173, 174, 179, 180, 191
\ref	21, 22	\tex_global:D	2083
\relax	21, 81	\tex_glueshrink:D	197, 892, 2168, 2182
\rmfamily	9, 10	\tex_gluestretch:D	196, 2166, 2180
\sbox	51	\tex_hbox:D	2084
\setbox	42, 50, 51	\tex_hskip:D	3206, 3207
\setCJKfallbackfamilyfont	11, 12	\tex_iffontchar:D	145
\setCJKfamilyfont	5, 9, 10	\tex_ignorespaces:D	3497, 3506
\setCJKmainfont	2, 9	\tex_italiccorrection:D	993, 995, 1026, 1028
\setCJKmathfont	11	\tex_kern:D	1287, 1288, 1985, 3538
\setCJKmonofont	9	\tex_lastkern:D	1088, 1104, 1154, 1255, 3410
\setCJKsansfont	9	\tex_lastnodetype:D	956, 1239, 2344
\sffamily	9, 10	\tex_lastskip:D	220, 904, 963, 1066, 1190, 1256, 1575, 1934, 2034, 2091, 2502, 3518
\shipout	24	\tex_nullfont:D	142
\spacefactor	26, 50	\tex_penalty:D	66, 68, 1181, 3513
\textbf	21	\tex_romannumeral:D	255
\textcolor	3, 21	\tex_setbox:D	2083
\textit	21	\tex_spacefactor:D	992, 1025, 2737
\textnormal	11	\tex_spaceskip:D	152, 158, 161, 184, 190, 196, 197
\textrm	9, 10	\tex_the:D	140, 647
\textsf	9, 10	\tex_unkern:D	1293
\texttt	9, 10	\tex_unskip:D	907, 964, 1069, 1191, 1296, 1578, 1935, 2100
\ttfamily	9, 10	\tex_XeTeXcharclass:D	272, 375, 585, 593, 598
\uline	19	\tex_XeTeXdashbreakstate:D	725
\ULon	19	\tex_XeTeXinterchartokenstate:D	329, 331
\ULthickness	18	\tex_XeTeXinterchartoks:D	641, 647, 695
\unskip	7	\tex_XeTeXrevision:D	371
\url	8	\tex_XeTeXversion:D	371
\verb	8, 9	\tex_xspaceskip:D	169, 177, 186, 194
\verbatim@font	9	textformat	18
\vrule	50	\texttt	3160
\xeCJK@document@hook	70, 73	thickness	18
\xeCJK@document@left@hook	71, 75	tl commands:	
\xeCJK@document@right@hook	72, 77	\c_space_tl	3263
\xeCJK@first@begindvi	91, 92	\tl_clear:N	2107, 2891, 2892
\xeCJKCancelSubCJKBlock	14	\tl_const:Nn	28, 97, 148, 709
\xeCJKchar	21	\tl_gclear:N	1298, 1828, 1969, 1971, 1976, 1978, 1980, 2016, 2067, 2297, 2373, 2400, 2530
\xeCJKDeclareCharClass	14	\tl_gput_right:Nn	80, 82, 84, 96
\xeCJKDeclarePunctStyle	5, 15	\tl_gset:Nn	1279, 1370, 1373, 1591, 1605, 1623, 1629, 1635, 1658, 1680, 1698, 1705,
\xeCJKDeclareSubCJKBlock	13, 14		
\xeCJKEditPunctStyle	15		
\xeCJKfntefbox	18, 19		
\xeCJKfntefon	19		
\xeCJKnobreak	6, 20		

1755, 1786, 1853, 1987, 1991, 1995, 1998, 2001, 2004, 2007, 2010, 2028, 2037, 2048, 2059, 2074, 2094, 2097, 2754	\token_if_math_toggle:NTF 3373
\tl_gset_eq:NN 1695, 2064, 2380	\token_if_space:NTF 999, 1032
\tl_head:n 1438, 1499	\token_to_meaning:N 274, 302
\tl_if_blank:nTF 561	\TrimSpaces 528
\tl_if_blank_p:n 558, 559	
\tl_if_empty:NTF 668, 672, 681, 1371, 1431, 1656, 1830, 1835, 1877, 1914, 1937, 2052, 2427, 2436, 2449, 2895, 2900	U
\tl_if_empty:NTF 278, 284, 290, 298, 1481, 3345	use commands:
\tl_if_eq:NNTF 1183, 1544	\use:N 527, 710, 1256, 1282, 1284, 2647, 3118
\tl_if_eq:NnTF 1375, 1379, 1598, 1660, 1664, 1710, 1739, 1767, 1803, 1866, 1923, 1943, 2024, 2046, 2054, 2378, 2483, 2550, 2581, 2584, 2596, 2599, 2614, 2617, 2645, 2659, 2686, 2693, 2701, 2971, 2979	\use_i:nn 863
\tl_if_exist:NTF 25, 1338, 1687	\use_none:n 231, 234, 235
\tl_if_head_eq_meaning:nNTF 1443, 1447, 1451, 1484, 1487, 1494, 3352	\use_none:nn 177, 194
\tl_if_head_is_group:nTF 1434, 1490	
\tl_if_head_is_N_type:nTF 121	V
\tl_if_novalue:nTF 565, 566	Verb 9
\tl_if_single:nTF 119	
\tl_if_single_token:nTF 2914	W
\tl_map_inline:nn 597	WidowPenalty 4
\tl_new:N 48, 49, 87, 88, 89, 110, 1275, 1320, 1321, 1322, 1323, 1329, 1330, 1331, 1332, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1415, 1416, 1690, 1692, 2306, 2365	
\tl_put_right:Nn 91, 109	X
\tl_replace_all:Nnn 683	xCJKecglue 3, 3218
\tl_reverse:N 1466	xeCJK commands:
\tl_set:Nn 666, 670, 679, 1429, 1464, 1492, 1541, 1542, 1676, 1807, 1812, 1827, 2112, 2113, 2114, 2115, 2116, 2119, 2123, 2124, 2132, 2265, 2266, 2310, 2368, 2379, 2390, 2395, 2415, 2416, 2431, 2432, 2439, 2472, 2670, 2674, 2676, 2678, 2680, 2708, 2719, 2721, 2893, 2917, 2920, 2926, 2935, 2936, 2937, 2938, 2939, 2944, 2954, 2955, 2956, 2995, 3005, 3026	\xeCJK_add_to_shipout:n 108, 108
\tl_set_eq:NN 1836, 2450	\xeCJK_allow_break: 65, 65
\tl_tail:n 1493	\xeCJK_app_inter_class_toks:nnn 658, 658, 663, 840, 3460
\tl_to_str:n 2943, 2960	\xeCJK_Boundary_and_Default: 839, 843, 843
\tl_trim_left_spaces:n 1493	\xeCJK_Boundary_and_FullLeft:N 3474
\tl_trim_right_spaces:n 1543	\xeCJK_Boundary_and_FullRight:N 3476
\tl_trim_spaces:n 115, 122, 125, 1430, 1465	\xeCJK_Boundary_and_NormalSp: 1007, 1008, 1008
\tl_use:N 74, 76, 78, 105, 1677, 1707, 1809, 1814, 2392, 2397, 2441, 2474, 2506, 2516, 2557, 2562, 2564, 2634, 2640, 2663, 2711	\xeCJK_check_for_ecglue: .. 846, 960, 3232, 3243, 3256
\tn 1474, 1475, 3099, 3103, 3155, 3156, 3157, 3159, 3161	\xeCJK_check_for_ecglue_normalsp: 1009, 1021, 3234, 3245, 3258
token commands:	\xeCJK_check_for_glue: 1040, 1047, 1047
\c_math_toggle_token 3289, 3402	\xeCJK_check_for_xglue: . 1176, 1185, 3231, 3242, 3255
\c_space_token 251	\xeCJK_CJK_and_Boundary:w 3286, 3287, 3287
\token_if_cs:NTF 2916	\xeCJK_CJK_and_CJK:N 3430, 3431, 3431
\token_if_eq_meaning:NNTF 3297, 3309, 3323, 3390	\xeCJK_class_group_begin: 720, 720, 823, 1041
\token_if_group_begin:NTF 3303, 3312, 3376	\xeCJK_class_group_end: 720, 728, 832, 3338, 3339, 3396, 3468, 3486, 3493, 3502
\token_if_group_end:NTF 3315	\xeCJK_class_num:n 526, 526, 547, 550, 591, 598, 641, 642, 647, 648, 696, 697
\token_if_letter:NTF 569	\xeCJK_clear_Boundary_and_CJK_toks: 688, 688, 692, 694, 826, 1042
\token_if_macro:NTF 3295, 3321, 3418	\xeCJK_clear_inter_class_toks:nn 650, 650, 673, 771, 772, 773, 774, 775, 825
	\xeCJK_copy_inter_class_toks:nnnn 664, 664, 734, 736, 738, 748, 749, 761, 762, 769, 783, 784, 787, 813, 814
	\xeCJK_cs_clear:N 128, 128, 3241, 3242
	\xeCJK_cs_gclear:N 128, 130
	\xeCJK_declare_char_class:nN 533, 536, 538, 606, 607, 608, 609, 627, 628, 629, 630, 631, 632, 634
	\xeCJK_declare_char_class:nn 530, 533, 533
	\xeCJK_declare_glue_node:n 1266, 1305
	\xeCJK_declare_node:n 1259, 1259, 1300, 1301, 1302, 1303, 1304, 1306, 1307, 1308
	\xeCJK_fallback_symbol:NN 795, 827, 1044, 3434, 3440, 3446

- \xeCJK_font_gset_to_current:N 139, 139
- \xeCJK_FullLeft_and_Boundary: 3478, 3481, 3481
- \xeCJK_FullLeft_and_CJK: 3439
- \xeCJK_FullRight_and_Boundary: 3480, 3499, 3499
- \xeCJK_FullRight_and_CJK: 3445
- \xeCJK_FullRight_and_CJStarter: 794, 799
- \xeCJK_get_inter_class_toks:nn 645, 645, 655, 661, 667, 671, 680
- \xeCJK_glue_to_skip:nN 213, 213, 1984, 1990, 1994, 3214, 3223, 3252, 3397
- \xeCJK_glyph_if_exist:N 143
- \xeCJK_glyph_if_exist:NTF 143
- \xeCJK_glyph_if_exist_p:N 143
- \xeCJK_if_CJK_class:N 304
- \xeCJK_if_CJK_class:NTF 304
- \xeCJK_if_CJK_class_p:N 304
- \xeCJK_if_last_node:n 1251
- \xeCJK_if_last_node:nTF 862, 864, 867, 870, 916, 919, 922, 925, 965, 968, 1018, 1192, 1195, 1207, 1210, 1213, 1251, 1554, 2129, 2139, 2142, 2150, 2264
- \xeCJK_if_last_node:TF 1893, 2108, 2191, 2195
- \xeCJK_if_last_node_p:n 1251
- \xeCJK_if_last_punct:TF 1174
- \xeCJK_if_package_loaded:n 23
- \xeCJK_if_package_loaded:nTF 23, 36
- \xeCJK_if_package_loaded_p:n 23
- \xeCJK_if_same_class:NN 313
- \xeCJK_if_same_class:NNTF 313
- \xeCJK_if_same_class_p:NN 313
- \xeCJK_ignore_spaces:w 3400, 3400
- \xeCJK_int_until_do:nn 229, 229, 573, 583
- \xeCJK_inter_class_toks:nnn 639, 639, 644, 651, 654, 660, 675, 684, 792, 820, 830, 838, 990, 1006, 1023, 1037, 3286, 3429, 3437, 3443, 3453, 3455, 3465, 3473, 3475, 3477, 3479
- \xeCJK_make_boundary: 319, 319
- \xeCJK_make_node:n 996, 1001, 1029, 1034, 1259, 1276, 2741, 2747, 2761, 3398, 3413
- \xeCJK_make_space_node: 1000, 1033, 3203, 3203, 3230, 3241, 3254, 3415
- \xeCJK_new_class:n 339, 339, 383, 384, 385, 388, 389, 390, 391, 394, 395, 396, 397, 398, 399
- \xeCJK_no_break: . 65, 67, 789, 791, 802, 3463, 3470, 3488
- \xeCJK_peek_catcode_ignore_spaces:NTF 238, 238, 3289, 3402
- \xeCJK_pre_inter_class_toks:nnn 652, 652, 657, 789, 790, 1004, 3529
- \xeCJK_punct_node:N 3487, 3494, 3503, 3542, 3542
- \xeCJK_remove_node: 865, 868, 871, 927, 942, 966, 973, 1019, 1056, 1092, 1108, 1114, 1117, 1157, 1159, 1161, 1163, 1165, 1193, 1196, 1208, 1211, 1214, 1290, 2131, 2152, 2155, 3413, 3415
- \xeCJK_replace_inter_class_toks:nnnn 677, 677
- \xeCJK_reset_space_factor: 209, 210, 212, 724
- \xeCJK_save_class:nn 352, 352, 369, 373, 374, 377, 378, 379
- \xeCJK_select_font: 824, 1043
- \xeCJK_set_char_class:nnn 546, 550, 579, 579
- \xeCJK_space_glue: 1404, 1985, 3240, 3263, 3273
- \xeCJK_space_or_xecglue: 1019, 1400, 3229, 3240, 3253, 3338, 3405, 3427
- \c_xeCJK_space_skip_tl 148, 886, 972, 1205, 2271
- \xeCJK_swap_cs:NN 132, 132
- \xeCJK_tl_remove_outer_braces:n 112, 112, 123
- \xeCJK_token_value_charcode:N 272, 273, 273
- \xeCJK_token_value_class:N 271, 271, 307, 315, 316
- \xeCJK_widow_penalty: 1118, 1161, 1402, 2595
- xeCJK internal commands:
 - _xeCJK_add_special_punct:nn 437
 - _xeCJK_after_end_preamble:n 70, 83
 - \g_xeCJK_after_end_preamble_hook_tl 78, 84, 89
 - _xeCJK_after_preamble:n 70, 81, 90
 - \g_xeCJK_after_preamble_hook_tl 74, 82, 88
 - \l_xeCJK_aligni_tl 1183
 - _xeCJK_at_end_preamble:n 70, 79
 - \g_xeCJK_at_end_preamble_hook_tl 76, 80, 87
 - \g_xeCJK_base_class_seq 700
 - \l_xeCJK_begin_int 236, 571, 573, 575, 576, 581, 583, 585, 586
 - \g_xeCJK_boundary_active_seq 1312, 1722, 1725, 1728, 1737, 1747, 1750, 1753, 1765, 1778, 1781, 1784, 1801, 1847, 1851, 1857, 1864
 - \l_xeCJK_boundary_active_tl 1332, 1738, 1739, 1766, 1767, 1802, 1803, 1865, 1866
 - _xeCJK_boundary_box_begin: 1310, 1729, 2071
 - _xeCJK_boundary_box_end:n 1310, 1740, 2282
 - _xeCJK_boundary_box_end_transparent:n 1310, 2437, 2456
 - _xeCJK_boundary_box_finish:nn 1310, 2285, 2294, 2388
 - _xeCJK_boundary_box_open:N 1310, 2077, 2081
 - _xeCJK_boundary_box_set_last_from_node: . . . 1310, 2366, 2408
 - _xeCJK_boundary_capture_allocate:n 1310, 1334, 1356, 1357, 1358, 1966
 - _xeCJK_boundary_capture_begin: 51, 1310, 1754, 1785, 1852, 1960, 2073
 - _xeCJK_boundary_capture_class:n 822, 833, 845, 1039, 1310, 1360, 1411, 1791, 1792
 - \g_xeCJK_boundary_capture_depth_int 954, 1310, 1363, 1368, 1594, 1601, 1608, 1618, 1626, 1632, 1638, 1649, 1654, 1756, 1787, 1810, 1815, 1839, 1854, 1869, 1962, 1964, 1967, 1970, 1973, 1977, 1979, 1981, 1988, 1992, 1996, 1999, 2002, 2005, 2008, 2012, 2018, 2031, 2040, 2049, 2055, 2060, 2065, 2075, 2078, 2095, 2098, 2293, 2300, 2313, 2371, 2393, 2398, 2442, 2448, 2453, 2461, 2465, 2477, 2509, 2519, 2537, 2654, 2689, 2696, 2704, 2714
 - _xeCJK_boundary_capture_emit_left:nn 1310, 1377, 1381, 1662, 1666, 1674
 - _xeCJK_boundary_capture_report_first:n 1310, 1646, 2433
 - _xeCJK_boundary_capture_resume: . 1310, 1703, 3202
 - _xeCJK_boundary_capture_space: . . 1310, 2021, 2087

`\_xeCJK_boundary_capture_suspend:` . 1310, 1684, 3200
`\_xeCJK_boundary_CJK_and_math:` . . . 3341, 3353, 3374
`\_xeCJK_boundary_emit_left:nn` 1310, 2444, 2534
`\_xeCJK_boundary_emit_left:nnn`
. 1310, 1678, 2536, 2540, 2629
`\_xeCJK_boundary_emit_left_math_space:nnn`
. 1310, 2543, 2546
`\_xeCJK_boundary_emit_left_normal:nnn`
. 1310, 2544, 2575
`\_xeCJK_boundary_enable_source_space:n`
. 1310, 2742, 2748, 2762, 2768
`\l\_xeCJK_boundary_first_tl` 1330,
1807, 1830, 1836, 2390, 2415, 2427, 2431, 2436, 2446, 2450
`\_xeCJK_boundary_group_end:n`
3305, 3318, 3326, 3328, 3349, 3357, 3381, 3386, 3389, 3393
`\_xeCJK_boundary_group_math:w` 3304, 3313, 3343, 3377
`\_xeCJK_boundary_hmode_transparent_begin:`
. 1310, 1843, 2811, 3133, 3148
`\_xeCJK_boundary_hmode_transparent_end:`
. 1310, 1861, 2813, 3135, 3150
`\_xeCJK_boundary_identity:n` 3298, 3324, 3364
`\_xeCJK_boundary_if_capture_box_sized:` 1310, 2307
`\_xeCJK_boundary_if_capture_box_sized:TF` . . 2337
`\_xeCJK_boundary_if_capture_box_visible:`
. 1310, 2334
`\_xeCJK_boundary_if_capture_box_visible:TF` . 2429
`\_xeCJK_boundary_if_last_math_node:` . . . 1310, 1549
`\_xeCJK_boundary_if_last_math_node:TF` . 1562, 1579
`\_xeCJK_boundary_if_last_math_space:` . . 1310, 2137
`\_xeCJK_boundary_if_last_math_space:TF`
. 873, 932, 977, 1078
`\_xeCJK_boundary_if_math_edge:n` 1310, 1510
`\_xeCJK_boundary_if_math_head:n` 1310, 1426
`\_xeCJK_boundary_if_math_head:nTF` . 1437, 1513, 1524
`\_xeCJK_boundary_if_math_tail:n` 1310, 1461
`\_xeCJK_boundary_if_math_tail:nTF` . 1498, 1516, 1530
`\_xeCJK_boundary_if_math_tail_reversed:n` . . 1479
`\_xeCJK_boundary_if_math_tail_reversed:nTF` . 1467
`\_xeCJK_boundary_if_math_tail_space:n` . 1310, 1538
`\_xeCJK_boundary_if_math_tail_space:nTF` . . . 1532
`\_xeCJK_boundary_if_registered:n` 1310, 2850
`\_xeCJK_boundary_if_registered:nTF` 3012, 3044
`\_xeCJK_boundary_inline_box_begin:` 1310, 1718, 2798
`\_xeCJK_boundary_inline_box_end:n` . 1310, 1734, 2800
`\_xeCJK_boundary_inline_last_box_begin:`
. 1310, 1743, 2785, 3171, 3183
`\_xeCJK_boundary_inline_last_box_end:n`
. 1310, 1762, 2787, 3175, 3187
`\_xeCJK_boundary_inline_stream_begin:n`
. 1310, 1771, 2834
`\_xeCJK_boundary_inline_stream_begin:nn`
. 1310, 1772, 1774
`\_xeCJK_boundary_inline_stream_end:n`
. 1310, 1798, 2836
`\_xeCJK_boundary_last_box_end:n` . . 1310, 1768, 2288
`\l\_xeCJK_boundary_last_tl` 1331,
1812, 1827, 1835, 1836, 1837, 2395, 2416, 2432, 2449,
2450, 2451, 2670, 2674, 2676, 2678, 2680, 2708, 2719, 2721
`\l\_xeCJK_boundary_math_actual_skip`
1418, 2043, 2164, 2166, 2168, 2169, 2180, 2182, 2203, 2257
`\_xeCJK_boundary_math_begin:` 1310, 1389, 1412
`\_xeCJK_boundary_math_begin:n` 1522, 3117, 3172, 3184
`\_xeCJK_boundary_math_begin_visible:`
. 1310, 1409, 1525
`\l\_xeCJK_boundary_math_delta_skip`
. 1420, 2235, 2252, 2277, 2278, 2567, 2570
`\l\_xeCJK_boundary_math_ecglue_dim`
. 1422, 2224, 2230, 2238, 2245, 2249, 2255
`\l\_xeCJK_boundary_math_ecglue_skip`
. 1419, 2225, 2237, 2246, 2254, 2273, 2563
`\_xeCJK_boundary_math_end:n`
. 1310, 1528, 3119, 3174, 3186
`\_xeCJK_boundary_math_end_confirm:`
. 1310, 1563, 1590, 1615
`\_xeCJK_boundary_math_end_confirm_node:`
. 1310, 1534, 1560
`\_xeCJK_boundary_math_end_confirm_space:`
. 1310, 1533, 1571
`\_xeCJK_boundary_math_end_confirm_space_`
`visible:` 1310, 1581, 1588
`\l\_xeCJK_boundary_math_shrink_dim`
. 1424, 2167, 2173, 2181, 2184, 2190, 2194, 2207
`\l\_xeCJK_boundary_math_skip`
. 1417, 1575, 1576, 1583, 1611, 2223, 2244, 2270, 2561
`\_xeCJK_boundary_math_space_consume:`
. 874, 933, 980, 1310, 2148, 2267
`\_xeCJK_boundary_math_space_decode:`
. 1310, 2120, 2153, 2187
`\l\_xeCJK_boundary_math_space_dim`
. 1421, 2222, 2231, 2243, 2250
`\_xeCJK_boundary_math_space_encode:` 1310, 2177, 2740
`\_xeCJK_boundary_math_space_expire:`
. 1310, 1924, 1949, 2528
`\_xeCJK_boundary_math_space_set_actual:n`
. 1310, 2162, 2514, 2555, 2709
`\_xeCJK_boundary_math_space_set_delta:n`
. 1310, 2218, 2274, 2565
`\_xeCJK_boundary_math_space_to_CJK:`
. 1081, 1167, 1169, 1310, 2262
`\l\_xeCJK_boundary_math_stretch_dim`
. 1423, 2165, 2172, 2179, 2183, 2189, 2198, 2206
`\l\_xeCJK_boundary_math_tl`
. 1415, 1429, 1431, 1435, 1438,
1444, 1448, 1452, 1464, 1466, 1468, 1492, 1495, 1541, 1544
`\l\_xeCJK_boundary_math_type_tl` 1416, 2265, 2266, 2275
`\l\_xeCJK_boundary_node_tl` 1329, 1394, 1395, 1542,
1544, 1676, 1679, 1806, 1822, 1913, 1914, 1923, 1927,
1936, 1937, 1943, 1954, 2023, 2024, 2046, 2052, 2066,
2377, 2378, 2379, 2381, 2439, 2445, 2472, 2481, 2483, 2488
`\_xeCJK_boundary_peek_math:w` 3299, 3325, 3365
`\_xeCJK_boundary_peek_math_branches:w` . 3368, 3371

_xeCJK_boundary_pop_node:N 1310, 1394, 1806, 1913, 1936, 2023, 2105, 2377
 \l_xeCJK_boundary_post_box 1328, 1881, 1884, 1885, 1886, 1890, 1918, 1922, 1940, 1948, 1952
 _xeCJK_boundary_post_transparent: 1310, 1873, 2824
 _xeCJK_boundary_post_transparent_relocate: 1310, 1889, 1911
 _xeCJK_boundary_post_transparent_relocate-glue: 1310, 1917, 1932
 _xeCJK_boundary_prepare_onearg_box:N 1310, 3104, 3192, 3193
 _xeCJK_boundary_prepare_onearg_box:NN 1310, 3110
 \l_xeCJK_boundary_probe_box 2305, 2340, 2346, 2374, 2383
 \g_xeCJK_boundary_probe_last_tl 2365, 2373, 2381, 2385
 \l_xeCJK_boundary_probe_name_tl 2306, 2310, 2318, 2324, 2328, 2342, 2368, 2376
 \g_xeCJK_boundary_probe_type_int . 2304, 2343, 2352
 _xeCJK_boundary_register_box:nn 1310, 2779, 3067, 3194, 3195
 _xeCJK_boundary_register_makeboxes: 1310, 3162, 3196
 _xeCJK_boundary_register_post_transparent:n 1310, 2817, 3083, 3197
 _xeCJK_boundary_register_stream:nn 1310, 2828, 3077
 _xeCJK_boundary_register_transparent:n 1310, 2804, 3081
 _xeCJK_boundary_register_wrapped_box:nn ... 1310, 2791, 3072
 \g_xeCJK_boundary_registered_prop .. 1313, 2781, 2783, 2793, 2796, 2806, 2809, 2819, 2822, 2830, 2832, 2852
 _xeCJK_boundary_replay_before: 1310, 1832, 1868, 2299, 2467, 2470
 _xeCJK_boundary_replay_before_math_space: .. 1310, 2484, 2498
 _xeCJK_boundary_replay_node:n 1310, 1837, 1896, 1927, 1954, 2451, 2485, 2488, 2523, 2725
 _xeCJK_boundary_reserve:N 1310, 2846, 3112, 3129, 3144
 _xeCJK_boundary_reserve:n 1310, 2844, 2848, 3165, 3198
 _xeCJK_boundary_reserve_space: 3300, 3348, 3356, 3380, 3385
 \g_xeCJK_boundary_reserved_prop .. 1314, 2845, 2855
 _xeCJK_boundary_restore_space: 1310, 2058, 2489, 2651
 _xeCJK_boundary_restore_space:n 1310, 2587, 2601, 2620, 2624, 2626, 2653, 2657
 _xeCJK_boundary_set_last_from_node:n 1310, 1822, 2385, 2667
 _xeCJK_boundary_set_math_tail_from_capture: 1310, 1823, 2409, 2684
 \g_xeCJK_boundary_suspend_depth_int 1311, 1366, 1621, 1652, 1686, 1688, 1691, 1693, 1696, 1699, 1708, 1711, 1715, 1721, 1746, 1777, 1846
 _xeCJK_boundary_use_ccglue:n 1310, 2602, 2604, 2637
 _xeCJK_boundary_use_ecglue:n 1310, 2586, 2589, 2606, 2619, 2622, 2631
 _xeCJK_boundary_use_glue:nn 1310, 2569, 2633, 2639, 2643, 2662
 \g_xeCJK_boundary_user_applied_seq 1316, 3051, 3089
 \l_xeCJK_boundary_user_invalid_bool 1318, 2889, 2898, 2903, 2905, 2907, 2921, 2927, 2945, 2961, 2974, 2983, 2993, 3003
 \l_xeCJK_boundary_user_mode_given_bool 1319, 2890, 2951, 2989, 2999
 \l_xeCJK_boundary_user_mode_tl 1322, 2893, 2954, 2955, 2956, 2971, 2979, 2992, 2995, 3002, 3005, 3029, 3062, 3068, 3073, 3078
 \l_xeCJK_boundary_user_name_tl 1320, 2891, 2895, 2917, 2920, 2926, 3013, 3016, 3020, 3023, 3033
 _xeCJK_boundary_user_register:n 1310, 2884
 _xeCJK_boundary_user_register_apply: . 1310, 3039
 _xeCJK_boundary_user_register_apply:nn 1310, 3050, 3056
 _xeCJK_boundary_user_register_check: . 1310, 3087
 \g_xeCJK_boundary_user_register_closed_bool . 1317, 2886, 3041
 _xeCJK_boundary_user_register_command:n ... 1310, 2912
 _xeCJK_boundary_user_register_mode:n . 1310, 2949
 _xeCJK_boundary_user_register_store: 1310, 2908, 3010
 _xeCJK_boundary_user_register_strategy:n ... 1310, 2931
 _xeCJK_boundary_user_register_validate: ... 1310, 2906, 2965
 \g_xeCJK_boundary_user_registered_prop 1315, 3020, 3032, 3042
 \l_xeCJK_boundary_user_strategy_tl 1321, 2892, 2900, 2935, 2936, 2937, 2938, 2939, 2944, 2967, 3028, 3060, 3063
 \l_xeCJK_boundary_user_value_seq 1324, 3058, 3060, 3062
 \l_xeCJK_boundary_user_value_tl .. 1323, 3026, 3034
 _xeCJK_boundary_wrap_transparent_noarg:NN .. 1310, 3125
 _xeCJK_boundary_wrap_transparent_onearg:NN . 1310, 3140
 _xeCJK_ccglue_or_space: 1114, 1159, 1193, 1208, 1225, 3272, 3278
 \l_xeCJK_ccglue_skip 3214, 3217
 _xeCJK_check_for_ecglue: ... 843, 852, 854, 960, 3243
 _xeCJK_check_for_ecglue_aux: 42, 843, 858, 860
 _xeCJK_check_for_ecglue_normalsp: 1014, 1016, 1022, 3246
 _xeCJK_check_for_glue_auxi: 1050, 1152
 _xeCJK_check_for_glue_auxii: 1059, 1144, 1150, 1172
 _xeCJK_check_for_glue_auxiii: 1175, 1178
 _xeCJK_check_for_glue_skip: 1053, 1064

_xeCJK_check_for_glue_skip_consume_and_- fallback:	1074, 1146	\g__xeCJK_HangulJamo_range_clist	622
_xeCJK_check_for_glue_skip_kern:	1071, 1076	\c__xeCJK_HangulJamoL_chars_clist	520, 630
_xeCJK_check_for_glue_skip_kern_normal: ...		\g__xeCJK_HangulJamoL_range_clist	623
.....	1084, 1086	\c__xeCJK_HangulJamoT_chars_clist	520, 632
_xeCJK_check_for_glue_skip_nonkern: ..	1072, 1129	\g__xeCJK_HangulJamoT_range_clist	625
_xeCJK_check_for_glue_skip_restore:		\c__xeCJK_HangulJamoV_chars_clist	520, 631
.....	1082, 1098, 1124, 1126, 1139, 1141	\g__xeCJK_HangulJamoV_range_clist	624
_xeCJK_check_for_xecglue:	848, 3232, 3256	\c__xeCJK_hyphens_chars_clist	427, 438, 461
_xeCJK_check_for_xecglue_normalsp: 1010, 3235, 3259		_xeCJK_if_last_glue:TF	217, 850, 856, 1012, 1052, 1188, 1227, 1295, 1573, 1916, 2026, 2089, 2500
_xeCJK_check_for_xglue:	1186, 3231, 3255	_xeCJK_if_last_hlist:TF	1227, 1879, 2290
_xeCJK_check_for_xglue_aux:	1197, 1201	_xeCJK_if_last_kern:TF	1049, 1070, 1227, 1292
_xeCJK_check_num_range:nnnn	555, 555, 581	_xeCJK_if_last_math:TF	1055, 1133, 1216, 1227, 1551
\c__xeCJK_CJ_chars_clist	444, 634	_xeCJK_if_last_none:TF	1227
\l__xeCJK_CJ_strict_bool	633, 637	_xeCJK_if_last_penalty:TF	1227
_xeCJK_CJK_and_Boundary_relax:N	3310, 3387	_xeCJK_info:nnn	56, 64
_xeCJK_CJK_and_math:	3290, 3335, 3342	_xeCJK_int_until_do:wn	229, 231, 234, 235
\c__xeCJK_CJK_chars_clist	463, 627	\l__xeCJK_interword_dim	880, 887, 889
\g__xeCJK_CJK_class_seq	700	\c__xeCJK_IS_chars_clist	443, 459
_xeCJK_CJK_class_tl:n	307, 311, 710	\c__xeCJK_iteration_marks_chars_clist	429
\l__xeCJK_CJK_group_bool	320, 723, 727	\l__xeCJK_last_kern_dim	2110, 2126, 2194, 2198
\g__xeCJK_CJK_range_clist	619	\g__xeCJK_last_node_tl	1275, 1279, 1298, 1697, 1705, 1828, 1877, 1896, 2067, 2297, 2400, 2530, 2754
\g__xeCJK_CJStarter_range_clist	626	\l__xeCJK_last_penalty_bool	1180
\c__xeCJK_CL_chars_clist	419, 456	\l__xeCJK_last_penalty_int	1181
\c__xeCJK_class_begin_int	380, 386	\g__xeCJK_last_punct_tl	801, 803, 3467, 3471, 3483, 3485, 3487, 3489, 3494, 3501, 3503, 3504
_xeCJK_class_csname:n		\l__xeCJK_last_skip	904, 905, 949, 963, 971, 974, 979, 981, 983, 987, 1066, 1067, 1096, 1110, 1120, 1137, 1143, 1182, 1190, 1204, 1218, 1223, 1934, 1939, 1947, 1955, 2091, 2092, 2096, 2502, 2504
.....	341, 345, 349, 354, 357, 362, 362, 363, 527, 710	\l__xeCJK_latin_punct_bool	612, 616
\g__xeCJK_class_seq		\c__xeCJK_left_tl	1183, 3489
.....	337, 347, 359, 701, 731, 744, 757, 779, 809	_xeCJK_make_node:N	
\c__xeCJK_CM_chars_clist	513, 629		1278, 1285, 2126, 2183, 2184, 3545, 3547
\g__xeCJK_CM_range_clist	621	_xeCJK_make_space_node:	2753, 3204, 3230, 3254
\l__xeCJK_ecglue_skip		_xeCJK_msg_new:nn	56, 56, 364, 2861, 2866, 2868, 2870, 2872, 2874, 2876, 2881
.....	865, 868, 943, 966, 973, 1056, 1095, 1136, 1163, 1165, 1196, 1211, 1214, 1217, 2273, 3223, 3252, 3264, 3397	_xeCJK_msg_new:nnn	57
\l__xeCJK_end_int	237, 572, 573, 581, 583	\g__xeCJK_new_class_seq	337, 348
_xeCJK_error:n	56, 58, 2887, 2919, 2925	\c__xeCJK_nobreak_penalty_int	68, 69
_xeCJK_error:nn	56, 59, 342, 355, 2897, 2902, 2942, 2959, 2973, 2981, 2991, 3001, 3015, 3022, 3046, 3093	_xeCJK_nobreak_zero_glue:	3495
\c__xeCJK_EX_chars_clist	441, 458	_xeCJK_node:n	1088, 1106, 1113, 1115, 1156, 1158, 1160, 1162, 1164, 1166, 1168, 1281, 2112, 2113, 2114, 2115, 2116, 2117, 2122, 2124, 3412, 3414
_xeCJK_font_gset_to_current_aux:NN	140, 141	\g__xeCJK_node_int ..	1261, 1264, 1268, 1271, 1273, 1274
\c__xeCJK_FullLeft_chars_clist	414, 608	\g__xeCJK_non_CJK_class_seq	700, 3449
\g__xeCJK_FullLeft_range_clist	604	\c__xeCJK_NormalSpace_chars_clist	400, 628
\c__xeCJK_FullRight_chars_clist	454, 609	\g__xeCJK_NormalSpace_range_clist	620
\g__xeCJK_FullRight_range_clist	605	\c__xeCJK_NS_chars_clist	427, 457
_xeCJK_get_charcode:w	274, 280, 286, 297	\c__xeCJK_OP_chars_clist	405, 416
_xeCJK_glue_check_expire_stale: 843, 875, 952, 1058		_xeCJK_orig_iframebox:w	3178, 3180, 3185
\g__xeCJK_glue_check_pending_bool		_xeCJK_orig_imakebox:w	3166, 3168, 3173
.....	42, 902, 929, 944, 945, 957, 1080, 1090, 1093, 1101, 1103, 1131, 1132, 1148, 1149, 1309, 1700, 1713, 1714, 1829, 2068, 2156, 2298, 2401, 2531, 2776, 3317	\c__xeCJK_package_ext_tl	25, 28
_xeCJK_glue_node:n	1283, 3206, 3207	_xeCJK_package_hook:nn	70, 85
\c__xeCJK_HalfLeft_chars_clist	400, 606		
\g__xeCJK_HalfLeft_range_clist	602		
\c__xeCJK_HalfRight_chars_clist	400, 607		
\l__xeCJK_halfright_nobreak_bool ..	3459, 3462, 3469		
\g__xeCJK_HalfRight_range_clist	603		

\g_xeCJK_par_guard_bool 3508, 3517, 3530, 3533, 3536, 3539
 \g_xeCJK_par_guard_dim 3509, 3518, 3538
 _xeCJK_peek_catcode_false:w 243, 263, 269
 _xeCJK_peek_catcode_ignore_spaces_branches:w 247, 249, 254
 _xeCJK_peek_catcode_true:w 241, 260, 268
 \l_xeCJK_peek_ignore_spaces_bool 27, 245, 252, 270, 3293, 3337, 3347, 3355, 3379, 3404, 3408
 \l_xeCJK_peek_search_token 240, 258, 267
 \c_xeCJK_PO_chars_clist 453, 460
 \l_xeCJK_pozzhao_ligature_bool 610, 615
 \c_xeCJK_PR_chars_clist 412, 417
 _xeCJK_punct_bound_unitization:NN 3544, 3549
 _xeCJK_punct_boundary_guard: 3490, 3505, 3508, 3510
 _xeCJK_punct_glue:NN 803, 3471, 3489, 3504
 _xeCJK_punct_if_middle:NTF 3483
 \g_xeCJK_punct_measure_fix_bool 3515
 _xeCJK_punct_rule:NN 801, 3467, 3485, 3501
 _xeCJK_recover_ecglue_source_space: 843, 857, 900
 _xeCJK_recover_ecglue_source_space_aux: ... 843, 908, 914
 _xeCJK_recover_ecglue_source_space_fallback: ... 843, 910, 912, 951
 _xeCJK_recover_ecglue_source_space_restore: ... 843, 928, 934, 947
 _xeCJK_recover_ecglue_source_space_success: ... 843, 917, 920, 923, 940
 _xeCJK_replace_space: 851, 961, 961, 1013
 _xeCJK_reserve_space_aux: 3419, 3424
 \l_xeCJK_reserve_space_bool 1109, 1119, 2003, 3271, 3277, 3285
 \c_xeCJK_right_tl 801, 803, 3467, 3471, 3485, 3501, 3504
 _xeCJK_save_CJK_class:n 706, 719
 _xeCJK_select_font: 804
 _xeCJK_set_char_class_aux:Nnw 533, 546, 552
 _xeCJK_set_char_class_eq:nn 589, 589
 \l_xeCJK_shipout_hook_bool 102, 104, 111
 \l_xeCJK_shipout_hook_tl 105, 109, 110
 _xeCJK_skip_if_interword:N 46, 843, 881
 _xeCJK_skip_if_interword:NTF 905, 979, 1067, 1576, 2092
 \g_xeCJK_space_factor_int 150, 163, 183, 204, 207, 209, 992, 1025, 2737
 _xeCJK_space_skip_scale:nnn . 165, 171, 184, 188, 201
 _xeCJK_swap_cs_aux:w 134, 136, 137
 _xeCJK_tl_remove_outer_braces:w 115, 117
 _xeCJK_tmp:nn 1228, 1243, 1244, 1245, 1246, 1247, 1248, 1249
 _xeCJK_tmp:w 276, 302, 435, 438, 439
 _xeCJK_tmp_aux:NNn 1230, 1234
 \l_xeCJK_tmp_bool 53
 \l_xeCJK_tmp_box 51, 216, 224
 \l_xeCJK_tmp_clist 55, 535, 536
 \l_xeCJK_tmp_dim 52, 885, 889, 2226, 2239, 2247, 2256, 3544, 3545, 3546, 3547
 \l_xeCJK_tmp_int 50, 363, 582, 585, 591, 593
 \l_xeCJK_tmp_skip 54, 886, 887, 1986, 1989, 1990, 1993, 1994, 1997
 \l_xeCJK_tmp_tl 48, 666, 668, 670, 672, 675, 679, 681, 683, 685
 \l_xeCJK_tmpb_tl 49
 _xeCJK_update_clear_toks:n 690, 712
 _xeCJK_warning:n 60
 _xeCJK_warning:nn 56, 61
 _xeCJK_warning:nnn 62
 _xeCJK_warning:nnnn 63
 \l_xeCJK_xecglue_bool 1094, 1135, 2006, 2268, 3228, 3239, 3250, 3265
 \c_xeCJK_xetex_version_str 370, 372
 xeCJKactive 3, 321
 \xeCJKCancelSubCJKBlock 14
 \xeCJKDeclareCharClass 14, 528
 \xeCJKDeclarePunctStyle 15
 \xeCJKDeclareSubCJKBlock 13
 \xeJKEditPunctStyle 15
 \xeCJKintefon 19
 \xeCJKnobreak 20
 \xeCJKOffVerbAddon 20
 \xeCJKResetCharClass 14, 617, 638
 \xeCJKResetPunctClass 14, 531, 600, 635
 \xeCJKRestoreSubCJKBlock 14
 \xeCJKsetkern 15
 \xeCJKsetup 2
 \xeCJKsetwidth 14
 \xeCJKShipoutHook 20, 90
 \xeCJKVerbAddon 20