

xeCJK 宏包

CTEX.ORG

2026/08/08 v3.10.5*

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	13	版本历史	231
3.5 标点符号的处理	14	代码索引	238
3.6 xeCJKfntef 用法说明	16		

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, 陆续由刘海洋¹、李清²及黄晨成³维护。目前活跃的维护者是黄晨成。

*[ctex-kit rev. 3ec789eb](#).

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 $\langle key \rangle = \langle val \rangle$ 的形式提供宏包选项, 用户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 { $\langle key_1 \rangle = \langle val_1 \rangle$ ,  $\langle key_2 \rangle = \langle val_2 \rangle$ , ...}
```

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 = {\textcolor{red}{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 = {\textcolor{red}{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 = {\textcolor{red}{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 = {\textcolor{red}{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 = {\textcolor{red}{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 10000; 也即, 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 $\backslash[\dots]$ 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- 的初始设置。

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

InlineEnv+ 使用 CheckSingle 选项时, xeCJK 会将 CJK 文字后接着的 $\text{\LaTeX}$ 环境起止点视为可断行点。
InlineEnv- 如果有某些特殊的 $\text{\LaTeX}$ 环境没有造成断行, 可以使用该选项声明, 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   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形。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 fwid 或 locl 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

```
\setCJKmainfont{Noto Serif CJK SC}[RawFeature+=fwid]
\xeCJKsetup{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

`\xeCJKsetup{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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 = <true|false>
```

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 10000`), 防止半角右标点出现在行首。

此选项的效果类似于 **LuaTeX-j**a 中的 `prebreakpenalty` 机制, 但受限于 **X_YTeX interchar** 机制, 粒度为字符级别而非逐字符级别。例如:

```
\xeCJKsetup{experiment/halfright-prebreakpenalty}
```

```
experiment/punct-measure-fix experiment/punct-measure-fix = <true|false>
```

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 `tabularray` 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:

```
\xeCJKsetup{experiment/punct-measure-fix}
```

```
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 **T_EX** 节点和命令实现的用户的实验性接口。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 **L^AT_EX3** 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]{\kernOpt #1\kernOpt}
\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 L^AT_EX 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。

特别地，命令本身就是一条赋值语句时不能用通用注册。\\global、\\long 这类前缀在 T_EX 中是“等待下一个赋值”的状态，而 before 钩子的代码插在该赋值之前执行；钩子里任何赋值都会把前缀消耗掉，调用方写的 \\global 便静默失效，且不产生任何错误或警告。\\sbox 属于这一类，本包为它使用专用适配器，把暂停观察移到盒子内部，使 \\global 始终紧邻 \\setbox (#1029)。用户注册自定义的“保存盒子”类命令时应同样处理。

顺带说明一个与本包无关的既有限制：\\global\\savebox 本来就不能跨分组生效。这是因为 \\savebox 是 robust 命令，\\global 在它自己的 \\@ifnextchar 前瞻阶段就被消耗，前缀根本到不了内部入口；不加载本包的原版 L^AT_EX 亦然。需要全局保存盒子时应使用 \\global\\sbox 或 \\global\\setbox。\\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。

注册的层级也会影响结果。框架在进入命令时就把 \\CJKecglue、\\CJKglue 和词间空格排入临时盒并读成数值，缓存的度量因此取决于进入命令那一刻生效的字体。如果目标命令的定义里包含字体切换，注册就必须包住最外层的那次切换：注册在切换内侧时，左边界会用切换后的字体度量，而右边界在命令结束、字体已经恢复之后求值，两侧数值不一致。这种写法在纯西文和纯中文文档里都看不出问题，只有中西文边界两侧都出现时才暴露。因此，若公开命令形如 \\newcommand\\mycmd[1]{\\texttt{...#1...}}，应当注册这个公开命令，而不是它内部负责排版参数的辅助命令 (#1046)。

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 = {true|false|env|env+}

true 表示在 \\verb 命令或 verbatim 环境里不自动调整中英文之间的间距。env 选项在 verbatim 环境里自动计算中西文间距和中文之间的间距，以便于保持代码的对齐；env 选项不调整 \\verb 里的间距，env+ 选项还将正文里设置的间距应用到 \\verb 里。这个选项对使用到 \\verbatim@font 命令的情形均有效，更一般的情况可以使用 3.7 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 = <true|false>`

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

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

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 = <{true|false|数字}>`
`AutoFakeSlant` `AutoFakeSlant = <{true|false|数字}>`

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

`Mapping` `Mapping = <{fullwidth-stop|full-stop|han-trad|han-simp|...}>`

`xeCJK` 提供了以上四个 TECKit 映射文件,可以在设置字体的时候通过 `Mapping` 选项来使用它们。其中 `fullwidth-stop` 用于将正常句号 `。` 转换成全角实心句号 `。`, `full-stop` 的作用相反。`han-trad` 用于将简体中文转换成繁体中文, `han-simp` 的作用相反。需要注意的是,简繁互换都是简单机械的字字对译,不能做到完全准确,使用时要小心。例如简体的“发挥”和“头发”被转换成繁体的“發揮”和“頭髮”,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 <{font name}> [<font features>]` 或
`\setCJKsansfont [<font features>] <{font name}>`

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

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

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

`\setCJKfamilyfont` ☆ `\setCJKfamilyfont <{family}> <{font name}> [<font features>]` 或
`\setCJKfamilyfont <{family}> [<font features>] <{font name}>`

声明新的 CJK 字体族 `<family>` 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 $\langle family \rangle$ 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 $\langle font features \rangle$ 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,Dólt,Corsivo,Cursief,kursywa,Itálico,Курсив,
Ítalik,Poševno,nghiêng,Etzana
Times New Roman:style=Negreta,cursiva,tučné kurzíva,fed kursiv,
Fett Kursiv,Εντονα Πλάγια,Bold Italic,Negrita Cursiva,
Lihavoitu Kursivoi,Gras Italique,Félkövé dólt,Grassetto Corsivo,
Vet Cursief,Halvfet Kursiv,Pogrubiona kursywa,Negrito Itálico,
Полужирный Курсив,Tučná kurzíva,Fet Kursiv,Kalín Ítalik,
Krepko poševno,nghiêng đậm,Lodi etzana
Times New Roman:style=Negreta,tučné,fed,Fett,Εντονα,Bold,Negrita,
Lihavoitu,Gras,Félkövé,Grassetto,Vet,Halvfet,Pogrubiona,Negrito,
Полужирный,Fet,Kalín,Krepko,đâm,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。

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。

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 }
```

会声明 SPUA 和 Ext-B 这两个子分区; 同时, 这将会在 3.2 节当中介绍的 CJK 字体设置命令的 里新建 SPUA 和 Ext-B 两个选项。新建的这两个选项, 其使用方法与 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

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, 同时设置 Ext-B 分区的字体是 SimSun-ExtB。

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

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

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

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

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

3.4 设置 CJK 字符范围

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

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

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

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

`\normalspacedchars` `\normalspacedchars {⟨char list⟩}`

在 `⟨char list⟩` 中出现的字符两端不自动添加空格, 初始设置是 `/`、`\` 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` ★ `\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 = (true|false)`: 是否使用 `\xeCJKsetup` 的 `PunctWidth`、`PunctBoundWidth` 选项和 `\xeCJKsetwidth`、`\xeCJKsetkern` 的设置。

`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/右空白宽度与当前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=<true|false>`: 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。

`add-min-bound-to-margin=<true|false>`: 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。

`optimize-margin=<true|false>`: 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。

`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=<true|false>`: 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。

`enabled-left-right-kerning=<true|false>`: 是否压缩左标点后紧跟右标点时二者之间的空白。预设值 `quanjiao` 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=<true|false>`: 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。

`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=<true|false>`: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。

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

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

3.6 xeCJKfntef 用法说明

`xeCJK` 包含有一个子宏包 `xeCJKfntef`, 可以用它来实现给汉字加点以及可正确断行的下划线等效果。它是 `CJKfntef` 宏包在 `XYLaTeX` 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:

`\CJKunderline` `\CJKunderline` [*] [-] [(选项)] {(内容)}

`\CJKunderdblline`

`\CJKunderwave` 虚室生白, 吉祥止止

`\CJKsout` 虚室生白, 吉祥止止

`\CJKxout` 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 \CJKunderline{虚室生白, 吉祥止止}\\
2 \CJKunderdblline{虚室生白, 吉祥止止}\\
3 \CJKunderwave{虚室生白, 吉祥止止}\\
4 \CJKsout{虚室生白, 吉祥止止}\\
5 \CJKxout{虚室生白, 吉祥止止}
```

```
1 \CJKunderline-{\南朝}\CJKunderline-{\梁}\CJKunderline-{\劉勰}%
```

```
2 \CJKunderwave-{\文心雕龍}\CJKunderwave-{\養氣}\\
```

```
3 \CJKunderline*[thickness=1pt, hidden=true]{瞻彼閔者, 虚室生白, 吉祥止止}
```

南朝梁劉勰文心雕龍養氣

`\CJKunderdot` `\CJKunderdot` [(选项)] {(内容)}

在汉字下加点, 可以和上述下划线命令嵌套使用, 嵌套后仍然可以正常断行。例如

虚室生白, 吉祥止止

虚室生白, 吉祥止止

```
1 \CJKunderline{虚室生白, \CJKunderdot{吉祥}止止}\\
2 \CJKunderdot{虚室生白, \CJKunderline{吉祥}止止}
```

下划线之间的嵌套则有断行方面的限制, 详见本节末尾“嵌套使用时的断行”。

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。

默认波浪线由 `l3draw` 绘制, 每个全角字符容纳四个完整周期, 符号的宽度、振幅和粗细都会随字号缩放。默认斜删除线也使用相同比例的矢量图案, 不再依赖数学字体。两种默认图案都用普通 `\leaders` 排列, 使同一行中被正文片段和中文字间距分开的图案保持相同相位。命令后的减号会让装饰在正文左右各内缩半个周期, 因此相邻命令之间会留下一个完整周期的断口。也可以用本选项换成其他波浪符号。例如, 下面改用 `lasy` 字体中的波浪符号:

`\xeCJKfntefon \xeCJKfntefon [*] [-] [{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{虚室生白, 吉祥止止}
```

虚室生白, 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。举例来说, 汉字 `\CJKsout{文字}` 汉字中 `\CJKsout` 前后的源码空格会被 `\xeCJK` 自动处理为 `CJK` 文字间距 (`CJKglue`); 间距方面的效果与汉字 文字 汉字一致。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。

3.6.1 嵌套使用时的断行

按装饰的绘制方式, `\xeCJKfntef` 的命令可以分成两类, 它们在嵌套时的断行能力不同:

- 线型命令: `\CJKunderline`、`\CJKunderdblline`、`\CJKunderwave`、`\CJKsout`、`\CJKxout` 和 `\CJKunderanyline`。它们借助 `ulem` 扫描正文并沿基线画出连续的线条。
- 符号型命令: `\CJKunderdot` 和 `\CJKunderanysymbol`。它们逐字放置独立的点或符号, 不需要 `ulem` 的扫描过程。

单独使用时, 以上命令都可以正常断行。符号型命令与任何命令嵌套, 也都保持正常断行。

但是两个线型命令相互嵌套时, 被嵌套的那一段无法断行。原因是 `ulem` 只允许最外层的线型命令启动扫描: 内层命令会复用外层已经打开的扫描过程, 它的正文被整段放进一个固定宽度的盒子, 盒子内部不再留有断点, 因此 `TEX` 只能让这一段溢出到页面右边, 并给出 `Overfull \hbox` 警告。嵌套的先后顺序不影响这一结果, 原生的 `\uline` 等 `ulem` 命令与本宏包的线型命令相互嵌套时同样如此。

这是 `ulem` 的扫描机制决定的, `\xeCJKfntef` 目前没有绕开它的办法。需要在较长的正文上同时得到两种装饰时, 可以改用下面的写法:

- 其中一种装饰改用符号型命令, 例如把内层的 `\CJKunderwave` 换成 `\CJKunderdot`;
- 按语义把正文分成若干段, 每段只用一个线型命令, 段与段之间保留断行位置;
- 只在确实不需要断行的短语上使用嵌套, 例如标题或表格单元内的少量文字。

此外, `\xeCJKfntef` 还提供了可指定宽度, 并让汉字两端分散对齐的环境 `CJKfilltwosides` 和 `CJKfilltwosides*`。

`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。 `CJKfilltwosides*` 环境与 `CJKfilltwosides` 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 阒 者,
虚 室 生 白, 吉 祥 止 止

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

瞻 彼 阒 者，
虚室生白，吉祥止止

```
1 \begin{CJKfilltwosides*}{0pt}
2   瞻彼阒者，\
3   虚室生白，吉祥止止
4 \end{CJKfilltwosides*}
```

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。

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

`\xeCJKKnobreak` 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 $\text{\TeX}$ 的输出例行程序（output routine）中的内容（比如页眉和页脚）。`\xeCJKShipoutHook` 用于恢复正文中的普通设置。`xeCJK` 已经处理了页眉和页脚的情况，其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， $\text{\TeX}$ 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 $\text{\TeX}$ 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 \foo{中} 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: `\CJKunderline`、`\CJKunderdblline`、`\CJKunderwave`、`\CJKsout`、`\CJKxout`、`\CJKunderdot`。
- 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 \CJKunderline{\textcolor{red}{中}} 后
前 \textcolor{red}{\CJKunderline{中}} 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, `xCJKecglue=false` 保留普通词间空格, `xCJKecglue=true` 则将其替换为 `CJKecglue`。

本节只讨论 CJK 间距,不涉及断行。xeCJKfntef 的线型命令相互嵌套时,间距仍然正确,但被嵌套的那一段无法断行,详见 3.6.1 节。

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。这即是说, 当用户在 `{...}`、`$...$` 或 `\begingroup...\endgroup` 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 `XYTeX` 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_p: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:Ne \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:nn
\__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:nn { \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:neeee { xeCJK } }
64 \cs_new_protected:Npn \__xeCJK_info:nee { \msg_info:nnee { xeCJK } }

```



```

\XeCJK_allow_break: 65 \cs_new_protected:Npn \XeCJK_allow_break:
\XeCJK_no_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
20 \AtBeginDocument { \XeCJK@document@hook }
21 \ctex_at_end_preamble:n { \XeCJK@document@left@hook }
22 \ctex_after_end_preamble:n { \XeCJK@document@right@hook }
23 \cs_new_protected:Npn \XeCJK@document@hook
24 { \tl_use:N \g__XeCJK_after_preamble_hook_tl }
25 \cs_new_protected:Npn \XeCJK@document@left@hook
26 { \tl_use:N \g__XeCJK_at_end_preamble_hook_tl }
27 \cs_new_protected:Npn \XeCJK@document@right@hook
28 { \tl_use:N \g__XeCJK_after_end_preamble_hook_tl }
29 \cs_new_protected:Npn \__XeCJK_at_end_preamble:n
30 { \tl_gput_right:Nn \g__XeCJK_at_end_preamble_hook_tl }
31 \cs_new_protected:Npn \__XeCJK_after_preamble:n
32 { \tl_gput_right:Nn \g__XeCJK_after_preamble_hook_tl }
33 \cs_new_protected:Npn \__XeCJK_after_end_preamble:n
34 { \tl_gput_right:Nn \g__XeCJK_after_end_preamble_hook_tl }
35 \cs_new_protected:Npn \__XeCJK_package_hook:nn
36 { \ctex_at_end_package:nn }
37 \tl_new:N \g__XeCJK_at_end_preamble_hook_tl
38 \tl_new:N \g__XeCJK_after_preamble_hook_tl
39 \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:nnTF \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:nnTF \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:nnTF \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 Xe_{La}TeX 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, Xe_{La}TeX 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 Xe_{La}TeX 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

```

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>

```

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   }
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:NTF
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:NNTF
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 Xe_{La}TeX 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

```

\xeCJK_new_class:n 新建一个字符类别。#1 是自定义名称。

```

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 }

```

\xeCJK_save_class:nn 保存 XeTeX 预定义的字符类别。#1 是自定义名称, #2 是编号。

```

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 }

```

__xeCJK_class_csname:n 字符类名称对应的控制序列名字。

```

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 $\text{\XeTeX}$ 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	〔	U+FE35	（
U+FE37	〰	U+FE39	〰	U+FE3B	〰	U+FE3D	〰	U+FE3F	〰	U+FE41	〰	U+FE43	〰
U+FE47	〰	U+FE59	（	U+FE5B	{	U+FE5D	{	U+FF08	（	U+FF3B	〔	U+FF5B	{
U+FF5F	（	U+FF62	〔										

以下代码的第一行是中西文共用的左引号。

```
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	〰	U+FE11	〰	U+FE12	〰	U+FE18	〰	U+FE36	〰	U+FE38	〰	U+FE3A	〰
U+FE3C	〰	U+FE3E	〰	U+FE40	〰	U+FE42	〰	U+FE44	〰	U+FE48	〰	U+FE50	,
U+FE52	.	U+FE5A	)	U+FE5C	}	U+FE5E	〕	U+FF09	)	U+FF0C	,	U+FF0E	.
U+FF3D	]]	U+FF5D	}	U+FF60	)	U+FF61	。〃	U+FF63	」	U+FF64	、		

以下代码的第一行是中西文共用的一些标点符号。

```
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+1B132	☒	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 , "1B132 , "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  }
```

\c_xeCJK_HangulJamoL_chars_clist 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
 \c_xeCJK_HangulJamoV_chars_clist 区分音节内 shaping 与相邻音节边界(#158)。
 \c_xeCJK_HangulJamoT_chars_clist

```
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 }

\xeCJK_declare_char_class:nn 用于设置字符所属的类别,#1 为类别名称,#2 为字符的 Unicode,相邻字符用半角逗号隔开,支
 \xeCJK_declare_char_class:nn 持类似"1100 -> "11FF 起止范围的使用方式。
 _xeCJK_set_char_class_aux:Nnw

```
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 }

```

\xeCJK_Boundary_and_Default: 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 }

```

这条路径补的是「源码空格 + 西文词」处的 \CJKecglue: 源码空格已被前视吃掉, marker 落在 ulem 刚打开的片段盒内部, 此时直接 \hskip 会把收缩量固化在定宽盒里, 行断行取不到它, 装饰段落因此溢出右边距 (#1037, 与 #1026 是同一机制的另一半: #1026 修的是西文词右侧, 这里是词前)。改经 __xeCJK_ulem_glue:n, 它在下划线状态下先 \UL@stop 关闭片段盒、把间距画成外层列表上的 \leaders、再 \UL@start 开新盒, 使收缩量回到行上; 不在下划线状态时它退化为普通 \hskip, 因此非 ulem 路径的行为不变。

不用 __xeCJK_boundary_use_ulem_glue:n: 它放的是裸 glue, 会在词前留下一段没有装饰线的空隙 (实测 300dpi 下断开 7px)。

默认实现是普通 \hskip; __xeCJK_ulem_glue:n 定义在 xeCJKfntef 里, 该宏包未加载时不存在, 所以由 xeCJKfntef 在加载时改写这个入口, 不能在这里直接引用。

```

860 \cs_new_protected:Npn \__xeCJK_use_ecglue_skip:
861 { \skip_horizontal:N \l__xeCJK_ecglue_skip }
862 \cs_new_protected:Npn \__xeCJK_check_for_ecglue_aux:
863 {
864   \xeCJK_if_last_node:nTF { CJK }
865   { \use_i:nn }
866   { \xeCJK_if_last_node:nTF { CJK-widow } }
867   { \xeCJK_remove_node: \__xeCJK_use_ecglue_skip: }
868   {
869     \xeCJK_if_last_node:nTF { CJK-space }
870     { \xeCJK_remove_node: \__xeCJK_use_ecglue_skip: }
871     {
872       \xeCJK_if_last_node:nTF { math }
873       { \xeCJK_remove_node: }
874       {
875         \__xeCJK_boundary_if_last_math_space:TF
876         { \__xeCJK_boundary_math_space_consume: }
877         { \__xeCJK_glue_check_expire_stale: }
878       }
879     }
880   }
881 }
882 \dim_new:N \l__xeCJK_interword_dim
883 \prg_new_conditional:Npnn \__xeCJK_skip_if_interword:N #1 { T , F , TF }
884 {
885   \skip_if_finite:nTF {#1}
886   {
887     \dim_set:Nn \l__xeCJK_tmp_dim {#1}
888     \skip_set:Nn \l__xeCJK_tmp_skip { \c_xeCJK_space_skip_tl }
889     \dim_set:Nn \l__xeCJK_interword_dim { \l__xeCJK_tmp_skip }
890     \dim_compare:nNnTF
891     { \l__xeCJK_tmp_dim } = { \l__xeCJK_interword_dim }

```

```

892         {
893             \dim_compare:nNnTF
894             { \tex_glueshrink:D #1 } > \c_zero_dim
895             { \prg_return_true: }
896             { \prg_return_false: }
897         }
898         { \prg_return_false: }
899     }
900     { \prg_return_false: }
901 }
902 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space:
903 {
904     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
905     {
906         \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
907         \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
908         {
909             \tex_unskip:D
910             \__xeCJK_recover_ecglue_source_space_aux:
911         }
912         { \__xeCJK_recover_ecglue_source_space_fallback: }
913     }
914     { \__xeCJK_recover_ecglue_source_space_fallback: }
915 }
916 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_aux:
917 {
918     \xeCJK_if_last_node:nTF { CJK }
919     { \__xeCJK_recover_ecglue_source_space_success: }
920     {
921         \xeCJK_if_last_node:nTF { CJK-widow }
922         { \__xeCJK_recover_ecglue_source_space_success: }
923         {
924             \xeCJK_if_last_node:nTF { CJK-space }
925             { \__xeCJK_recover_ecglue_source_space_success: }
926             {
927                 \xeCJK_if_last_node:nTF { math }
928                 {
929                     \xeCJK_remove_node:
930                     \__xeCJK_recover_ecglue_source_space_restore:
931                     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
932                 }
933                 {
934                     \__xeCJK_boundary_if_last_math_space:TF
935                     { \__xeCJK_boundary_math_space_consume: }
936                     { \__xeCJK_recover_ecglue_source_space_restore: }
937                 }
938             }
939         }
940     }
941 }
942 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_success:
943 {
944     \xeCJK_remove_node:
945     \__xeCJK_use_ecglue_skip:
946     \bool_if:NT \g__xeCJK_glue_check_pending_bool
947     { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
948 }
949 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_restore:
950 {
951     \skip_horizontal:N \l__xeCJK_last_skip
952 }
953 \cs_new_eq:NN \__xeCJK_recover_ecglue_source_space_fallback: \prg_do_nothing:
954 \cs_new_protected:Npn \__xeCJK_glue_check_expire_stale:
955 {
956     \int_compare:nNnT { \g__xeCJK_boundary_capture_depth_int } = \c_zero_int
957     {
958         \int_compare:nNnT { \tex_lastnodetype:D } < \c_zero_int

```

```

959         { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
960     }
961 }
962 \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 L^AT_EX 2_ε 中常用的 \hrulefill 和 \dotfill 定义末尾都有 \kern\z@ 保护。

```

963 \cs_new_protected:Npn \__xeCJK_replace_space:
964 {
965     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
966     \tex_unskip:D
967     \xeCJK_if_last_node:nTF { CJK-space }
968     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
969     {
970         \xeCJK_if_last_node:nTF { CJK }
971         {
972             \skip_if_eq:nnTF
973             { \l__xeCJK_last_skip }
974             { \c_xeCJK_space_skip_tl }
975             { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
976             { \skip_horizontal:N \l__xeCJK_last_skip }
977         }
978         {
979             \__xeCJK_boundary_if_last_math_space:TF
980             {
981                 \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
982                 { \__xeCJK_boundary_math_space_consume: }
983                 { \skip_horizontal:N \l__xeCJK_last_skip }
984             }
985             { \skip_horizontal:N \l__xeCJK_last_skip }
986         }
987     }
988 }
989 \skip_new:N \l__xeCJK_last_skip

990 \clist_map_inline:nn { Default , HalfRight }
991 {
992     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
993     {
994         \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
995         \peek_meaning_remove:NTF \tex_italiccorrection:D
996         {
997             \tex_italiccorrection:D
998             { \xeCJK_make_node:n { default } }
999         }
1000         {
1001             \token_if_space:NTF \l_peek_token
1002             { \xeCJK_make_space_node: }
1003             { { \xeCJK_make_node:n { default } } }
1004         }
1005     }
1006     \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue }
1007 }

1008 \xeCJK_inter_class_toks:nnn { Boundary } { NormalSpace }
1009 { \xeCJK_Boundary_and_NormalSp: }

\__xeCJK_Boundary_and_NormalSp: 1010 \cs_new_protected:Npn \xeCJK_Boundary_and_NormalSp:
1011 { \xeCJK_check_for_ecglue_normalsp: }
1012 \cs_new_protected:Npn \__xeCJK_check_for_xecglue_normalsp:
1013 {
1014     \__xeCJK_if_last_glue:TF
1015     { \__xeCJK_replace_space: }
1016     { \__xeCJK_check_for_ecglue_normalsp: }
1017 }
1018 \cs_new_protected:Npn \__xeCJK_check_for_ecglue_normalsp:

```

```

1019 {
1020   \xeCJK_if_last_node:nT { CJK-space }
1021   { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
1022 }
1023 \cs_new_eq:NN \xeCJK_check_for_ecglue_normalsp:
1024   \__xeCJK_check_for_ecglue_normalsp:

1025 \xeCJK_inter_class_toks:nnn { NormalSpace } { Boundary }
1026 {
1027   \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
1028   \peek_meaning_remove:NTF \tex_italiccorrection:D
1029   {
1030     \tex_italiccorrection:D
1031     { \xeCJK_make_node:n { normalspace } }
1032   }
1033   {
1034     \token_if_space:NTF \l_peek_token
1035     { \xeCJK_make_space_node: }
1036     { { \xeCJK_make_node:n { normalspace } } }
1037   }
1038 }

1039 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
1040 {
1041   \__xeCJK_boundary_capture_class:n { CJK }
1042   \xeCJK_check_for_glue:
1043   \xeCJK_class_group_begin:
1044   \xeCJK_clear_Boundary_and_CJK_toks:
1045   \xeCJK_select_font:
1046   \xeCJK_fallback_symbol:NN
1047   \CJKsymbol
1048 }

\xeCJK_check_for_glue: 1049 \cs_new_protected:Npn \xeCJK_check_for_glue:
1050 {
1051   \__xeCJK_if_last_kern:TF
1052   { \__xeCJK_check_for_glue_auxi: }
1053   {
1054     \__xeCJK_if_last_glue:TF
1055     { \__xeCJK_check_for_glue_skip: }
1056     {
1057       \__xeCJK_if_last_math:TF
1058       { \xeCJK_remove_node: \__xeCJK_use_ecglue_skip: }
1059       {
1060         \__xeCJK_glue_check_expire_stale:
1061         \__xeCJK_check_for_glue_auxii:
1062       }
1063     }
1064   }
1065 }

```

统一 framework 在已知命令出口重放 *xeCJK* marker, 并通过 `\g__xeCJK_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__xeCJK_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 `\__xeCJK_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。

```

1066 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip:
1067 {

```

```

1068 \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1069 \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
1070 {
1071   \tex_unskip:D
1072   \__xeCJK_if_last_kern:TF
1073   { \__xeCJK_check_for_glue_skip_kern: }
1074   { \__xeCJK_check_for_glue_skip_nonkern: }
1075 }
1076 { \__xeCJK_check_for_glue_skip_consume_and_fallback: }
1077 }
1078 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_kern:
1079 {
1080   \__xeCJK_boundary_if_last_math_space:TF
1081   {
1082     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1083     { \__xeCJK_boundary_math_space_to_CJK: }
1084     { \__xeCJK_check_for_glue_skip_restore: }
1085   }
1086   { \__xeCJK_check_for_glue_skip_kern_normal: }
1087 }
1088 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_kern_normal:
1089 {
1090   \dim_compare:nNnTF \tex_lastkern:D = { \__xeCJK_node:n { math } }
1091   {
1092     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1093     {
1094       \xeCJK_remove_node:
1095       \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1096       \bool_if:NTF \l__xeCJK_xecglue_bool
1097       { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1098       { \skip_horizontal:N \l__xeCJK_last_skip }
1099     }
1100     { \__xeCJK_check_for_glue_skip_restore: }
1101   }
1102   {
1103     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1104     {
1105       \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1106       \dim_case:nnF { \tex_lastkern:D }
1107       {
1108         { \__xeCJK_node:n { CJK } }
1109         {
1110           \xeCJK_remove_node:
1111           \bool_if:NTF \l__xeCJK_reserve_space_bool
1112           { \skip_horizontal:N \l__xeCJK_last_skip }
1113           { \CJKglue }
1114         }
1115         { \__xeCJK_node:n { CJK-space } }
1116         { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1117         { \__xeCJK_node:n { CJK-widow } }
1118         {
1119           \xeCJK_remove_node:
1120           \xeCJK_widow_penalty:
1121           \bool_if:NTF \l__xeCJK_reserve_space_bool
1122           { \skip_horizontal:N \l__xeCJK_last_skip }
1123           { \CJKglue }
1124         }
1125       }
1126       { \__xeCJK_check_for_glue_skip_restore: }
1127     }
1128     { \__xeCJK_check_for_glue_skip_restore: }
1129   }
1130 }
1131 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_nonkern:
1132 {
1133   \bool_if:NT \g__xeCJK_glue_check_pending_bool
1134   { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }

```



```

1135 \__xeCJK_if_last_math:TF
1136 {
1137     \bool_if:NTF \l__xeCJK_xecglue_bool
1138     { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1139     { \skip_horizontal:N \l__xeCJK_last_skip }
1140 }
1141 { \__xeCJK_check_for_glue_skip_restore: }
1142 }
1143 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_restore:
1144 {
1145     \skip_horizontal:N \l__xeCJK_last_skip
1146     \__xeCJK_check_for_glue_auxii:
1147 }
1148 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_consume_and_fallback:
1149 {
1150     \bool_if:NT \g__xeCJK_glue_check_pending_bool
1151     { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
1152     \__xeCJK_check_for_glue_auxii:
1153 }
1154 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxi:
1155 {
1156     \dim_case:nn { \tex_lastkern:D }
1157     {
1158         { \__xeCJK_node:n { CJK } }
1159         { \xeCJK_remove_node: \CJKglue }
1160         { \__xeCJK_node:n { CJK-space } }
1161         { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1162         { \__xeCJK_node:n { CJK-widow } }
1163         { \xeCJK_remove_node: \xeCJK_widow_penalty: \CJKglue }
1164         { \__xeCJK_node:n { default } }
1165         { \xeCJK_remove_node: \__xeCJK_use_ecglue_skip: }
1166         { \__xeCJK_node:n { math } }
1167         { \xeCJK_remove_node: \__xeCJK_use_ecglue_skip: }
1168         { \__xeCJK_node:n { math-space } }
1169         { \__xeCJK_boundary_math_space_to_CJK: }
1170         { \__xeCJK_node:n { math-space-frozen } }
1171         { \__xeCJK_boundary_math_space_to_CJK: }
1172     }
1173 }
1174 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxii:
1175 {
1176     \xeCJK_if_last_punct:TF
1177     { \__xeCJK_check_for_glue_auxiii: }
1178     { \xeCJK_check_for_xglue: }
1179 }
1180 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxiii:
1181 {
1182     \bool_if:NT \l__xeCJK_last_penalty_bool
1183     { \tex_penalty:D \l__xeCJK_last_penalty_int }
1184     \skip_horizontal:N \l__xeCJK_last_skip
1185     \tl_if_eq:NNF \l__xeCJK_aligni_tl \c__xeCJK_left_tl { \CJKglue }
1186 }
1187 \cs_new_eq:NN \xeCJK_check_for_xglue: \prg_do_nothing:
1188 \cs_new_protected:Npn \__xeCJK_check_for_xglue:
1189 {
1190     \__xeCJK_if_last_glue:T
1191     {
1192         \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1193         \tex_unskip:D
1194         \xeCJK_if_last_node:nTF { CJK-space }
1195         { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1196         {
1197             \xeCJK_if_last_node:nTF { default-space }
1198             { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1199             { \__xeCJK_check_for_xglue_aux: }
1200         }
1201     }

```

```

1202 }
1203 \cs_new_protected:Npn \__xeCJK_check_for_xglue_aux:
1204 {
1205   \skip_if_eq:nnTF
1206     { \l__xeCJK_last_skip }
1207     { \c_xeCJK_space_skip_tl }
1208   {
1209     \xeCJK_if_last_node:nTF { CJK }
1210     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1211     {
1212       \xeCJK_if_last_node:nTF { default }
1213       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1214       {
1215         \xeCJK_if_last_node:nTF { math }
1216         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1217         {
1218           \__xeCJK_if_last_math:TF
1219           { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1220           { \skip_horizontal:N \l__xeCJK_last_skip }
1221         }
1222       }
1223     }
1224   }
1225   { \skip_horizontal:N \l__xeCJK_last_skip }
1226 }
1227 \cs_new_protected:Npn \__xeCJK_ccglue_or_space:
1228 { \CJKglue }

```

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

__xeCJK_if_last_hlist:TF
__xeCJK_if_last_math:TF
__xeCJK_if_last_glue:TF
__xeCJK_if_last_kern:TF
__xeCJK_if_last_penalty:TF

```

1229 \group_begin:
1230 \cs_set:Npn \__xeCJK_tmp:nn #1
1231 {
1232   \exp_args:Ncc \__xeCJK_tmp_aux:NNn
1233     { __xeCJK_if_last_ #1 : }
1234     { c__xeCJK_ #1_node }
1235 }
1236 \cs_set:Npn \__xeCJK_tmp_aux:NNn #1#2#3
1237 {
1238   \int_const:Nn #2 {#3}
1239   \prg_new_conditional:Npnn #1 { T , F , TF }
1240   {
1241     \if_int_compare:w \tex_lastnodetype:D = #2
1242       \prg_return_true: \else: \prg_return_false: \fi:
1243   }
1244 }
1245 \__xeCJK_tmp:nn { none } { -1 }
1246 \__xeCJK_tmp:nn { hlist } { 1 }
1247 \__xeCJK_tmp:nn { math } { 10 }
1248 \__xeCJK_tmp:nn { glue } { 11 }
1249 \__xeCJK_tmp:nn { kern } { 12 }
1250 \__xeCJK_tmp:nn { penalty } { 13 }
1251 \__xeCJK_tmp:nn { whatsit } { 9 }
1252 \group_end:

```

```

\xeCJK_if_last_node_p:n 1253 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
\xeCJK_if_last_node:nTF 1254 {
1255   \if_dim:w
1256     \cs_if_exist_use:cTF { c__xeCJK_#1_node_dim }
1257     { = \tex_lastkern:D }
1258     { \use:c { c__xeCJK_#1_node_skip } = \tex_lastskip:D }
1259     \prg_return_true: \else: \prg_return_false: \fi:
1260 }

```

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

\xeCJK_make_node:n

```

1261 \cs_new_protected:Npn \xeCJK_declare_node:n #1

```

```

1262 {
1263   \int_gincr:N \g__xeCJK_node_int
1264   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
1265   { \dim_gset:cn } { \dim_const:cn }
1266   { c__xeCJK_#1_node_dim } { \g__xeCJK_node_int sp }
1267 }
1268 \cs_new_protected:Npn \xeCJK_declare_glue_node:n #1
1269 {
1270   \int_gincr:N \g__xeCJK_node_int
1271   \skip_if_exist:cTF { c__xeCJK_#1_node_skip }
1272   { \skip_gset:cn } { \skip_const:cn }
1273   { c__xeCJK_#1_node_skip } { \g__xeCJK_node_int sp }
1274 }
1275 \int_new:N \g__xeCJK_node_int
1276 \int_gset:Nn \g__xeCJK_node_int { 10 }
1277 \tl_new:N \g__xeCJK_last_node_tl
1278 \cs_new_protected:Npn \xeCJK_make_node:n #1
1279 {
1280   \exp_args:Nc \__xeCJK_make_node:N { c__xeCJK_#1_node_dim }
1281   \tl_gset:Nn \g__xeCJK_last_node_tl {#1}
1282 }
1283 \cs_new:Npn \__xeCJK_node:n #1
1284 { \use:c { c__xeCJK_#1_node_dim } }
1285 \cs_new:Npn \__xeCJK_glue_node:n #1
1286 { \use:c { c__xeCJK_#1_node_skip } }
1287 \cs_new_protected:Npn \__xeCJK_make_node:N #1
1288 {
1289   \tex_kern:D - #1
1290   \tex_kern:D #1
1291 }
1292 \cs_new_protected:Npn \xeCJK_remove_node:
1293 {
1294   \__xeCJK_if_last_kern:TF
1295   { \tex_unkern:D \tex_unkern:D }
1296   {
1297     \__xeCJK_if_last_glue:T
1298     { \tex_unskip:D \tex_unskip:D }
1299   }
1300   \tl_gclear:N \g__xeCJK_last_node_tl
1301 }
1302 \xeCJK_declare_node:n { CJK }
1303 \xeCJK_declare_node:n { CJK-space }
1304 \xeCJK_declare_node:n { default }
1305 \xeCJK_declare_node:n { CJK-widow }
1306 \xeCJK_declare_node:n { normalspace }
1307 \xeCJK_declare_glue_node:n { default-space }
1308 \xeCJK_declare_node:n { math }
1309 \xeCJK_declare_node:n { math-space }
1310 \xeCJK_declare_node:n { math-space-frozen }
1311 \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, 不把事件广播到外层 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。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。

```

1312 \int_new:N \g__xeCJK_boundary_capture_depth_int
1313 \int_new:N \g__xeCJK_boundary_suspend_depth_int
1314 \seq_new:N \g__xeCJK_boundary_active_seq
1315 \prop_new:N \g__xeCJK_boundary_registered_prop
1316 \prop_new:N \g__xeCJK_boundary_reserved_prop
1317 \prop_new:N \g__xeCJK_boundary_user_registered_prop
1318 \seq_new:N \g__xeCJK_boundary_user_applied_seq
1319 \bool_new:N \g__xeCJK_boundary_user_register_closed_bool
1320 \bool_new:N \l__xeCJK_boundary_user_invalid_bool
1321 \bool_new:N \l__xeCJK_boundary_user_mode_given_bool
1322 \tl_new:N \l__xeCJK_boundary_user_name_tl
1323 \tl_new:N \l__xeCJK_boundary_user_strategy_tl
1324 \tl_new:N \l__xeCJK_boundary_user_mode_tl
1325 \tl_new:N \l__xeCJK_boundary_user_value_tl
1326 \seq_new:N \l__xeCJK_boundary_user_value_seq
1327 \box_new:c { g__xeCJK_boundary_capture_1_box }
1328 \box_new:c { g__xeCJK_boundary_capture_2_box }
1329 \box_new:c { g__xeCJK_boundary_capture_3_box }
1330 \box_new:N \l__xeCJK_boundary_post_box
1331 \tl_new:N \l__xeCJK_boundary_node_tl
1332 \tl_new:N \l__xeCJK_boundary_first_tl
1333 \tl_new:N \l__xeCJK_boundary_last_tl
1334 \tl_new:N \l__xeCJK_boundary_active_tl
1335
1336 \cs_new_protected:Npn \__xeCJK_boundary_capture_allocate:n #1
1337 {
1338   \box_if_exist:cF { g__xeCJK_boundary_capture_#1_box }
1339   { \box_new:c { g__xeCJK_boundary_capture_#1_box } }
1340   \tl_if_exist:cF { g__xeCJK_boundary_capture_#1_before_tl }
1341   {
1342     \tl_new:c { g__xeCJK_boundary_capture_#1_before_tl }
1343     \tl_new:c { g__xeCJK_boundary_capture_#1_before_math_space_skip_tl }
1344     \tl_new:c { g__xeCJK_boundary_capture_#1_first_tl }
1345     \tl_new:c { g__xeCJK_boundary_capture_#1_last_tl }
1346     \tl_new:c { g__xeCJK_boundary_capture_#1_kind_tl }
1347     \tl_new:c { g__xeCJK_boundary_capture_#1_space_skip_tl }
1348     \tl_new:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl }
1349     \tl_new:c { g__xeCJK_boundary_capture_#1_ccglue_skip_tl }
1350     \tl_new:c { g__xeCJK_boundary_capture_#1_space_flag_tl }
1351     \tl_new:c { g__xeCJK_boundary_capture_#1_reserve_space_flag_tl }
1352     \tl_new:c { g__xeCJK_boundary_capture_#1_xecglue_flag_tl }
1353     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_flag_tl }
1354     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_space_flag_tl }
1355     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_space_skip_tl }
1356   }
1357 }
1358 \__xeCJK_boundary_capture_allocate:n { 1 }
1359 \__xeCJK_boundary_capture_allocate:n { 2 }
1360 \__xeCJK_boundary_capture_allocate:n { 3 }
1361
1362 \cs_new_protected:Npn \__xeCJK_boundary_capture_class:n #1
1363 {
1364   \int_compare:nNtT
1365   { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1366   {
1367     \int_compare:nNtT
1368     { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1369     {
1370       \int_step_inline:nn { \g__xeCJK_boundary_capture_depth_int }

```

```

1371         {
1372             \tl_gset:cn { g__xeCJK_boundary_capture_###1_last_tl } {#1}
1373             \tl_if_empty:cT { g__xeCJK_boundary_capture_###1_first_tl }
1374                 {
1375                     \tl_gset:cn
1376                     { g__xeCJK_boundary_capture_###1_first_tl } {#1}
1377                     \tl_if_eq:cnTF
1378                     { g__xeCJK_boundary_capture_###1_kind_tl } { stream }
1379                     { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1380                     {
1381                         \tl_if_eq:cnT
1382                         { g__xeCJK_boundary_capture_###1_kind_tl } { stream-ulem }
1383                         { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1384                     }
1385                 }
1386             }
1387         }
1388     }
1389 }
1390
1391 \cs_new_protected:Npn \__xeCJK_boundary_math_begin:
1392 {
1393     \mode_if_horizontal:T
1394     {
1395         \scan_stop:
1396         \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1397         \str_case:Vn \l__xeCJK_boundary_node_tl
1398         {
1399             { CJK }
1400             { \CJKecglue }
1401             { CJK-space }
1402             { \xeCJK_space_or_xecglue: }
1403             { CJK-widow }
1404             { \xeCJK_widow_penalty: \CJKecglue }
1405             { default-space }
1406             { \xeCJK_space_glue: }
1407         }
1408     }
1409 }
1410
1411 \cs_new_protected:Npn \__xeCJK_boundary_math_begin_visible:
1412 {
1413     \__xeCJK_boundary_capture_class:n { math }
1414     \__xeCJK_boundary_math_begin:
1415 }
1416
1417 \tl_new:N \l__xeCJK_boundary_math_tl
1418 \tl_new:N \l__xeCJK_boundary_math_type_tl
1419 \skip_new:N \l__xeCJK_boundary_math_skip
1420 \skip_new:N \l__xeCJK_boundary_math_actual_skip
1421 \skip_new:N \l__xeCJK_boundary_math_ecglue_skip
1422 \skip_new:N \l__xeCJK_boundary_math_delta_skip
1423 \dim_new:N \l__xeCJK_boundary_math_space_dim
1424 \dim_new:N \l__xeCJK_boundary_math_ecglue_dim
1425 \dim_new:N \l__xeCJK_boundary_math_stretch_dim
1426 \dim_new:N \l__xeCJK_boundary_math_shrink_dim
1427
1428 % \changes{v3.10.5}{2026/08/03}{boundary 判断前把参数副本中 catcode 4 的
1429 %   |&| 换成 \cs{scan_stop:}, 修复 \tn{colorbox} 等命令在 \tn{halign}
1430 %   语境下参数含 |&| 时编译报错的问题 (\#1043)。}
1431 %
1432 % \cs{__xeCJK_boundary_math_set:n} 把待判断的参数存入 \cs{l__xeCJK_boundary_math_tl},
1433 % 并把其中 catcode 4 的 |&| 换成 \cs{scan_stop:}。
1434 %
1435 % 触发条件要分清两件事: |&| 在 \LaTeX{} 下\emph{默认}就是 catcode 4
1436 %   (\file{latex.ltx}), 并非 \tn{halign} 把它设成 4; 真正要紧的是\textbf{扫描发生的
1437 %   位置}——当 \cs{tl_if_head_eq_meaning:nNTF} 等 \pkg{expl3} 条件式在\emph{对齐环境内}

```

```

1438 % (\env{eqnarray}、\env{align}、\env{tabular}) 读取参数时, catcode 4 的 |&| 会终止
1439 % 它正在读的宏参数, 报
1440 % \enquote{Argument of \cs{__tl_tl_head:w} has an extra \}} (\#1043)。
1441 % 同一个 token list 在对齐环境\emph{之外}走同一条条件式实测 0 错误, 所以这不是
1442 % 「\pkg{expl3} 条件式对 catcode-4 记号的固有限制」, 而是对齐符在对齐环境里的
1443 % 参数终止语义。
1444 % 这里只是为\emph{语法判断}准备副本, 实际排版仍使用原始参数, 所以替换不影响
1445 % 输出; 换成 \cs{scan_stop:} 而不是直接删除, 是为了保住 |&| 的位置,
1446 % 使 |&$x$| 这类参数仍被正确判定为「首项不是公式」。
1447 %
1448 % 两条范围说明, 避免后人误以为漏改:
1449 % \begin{itemize}
1450 %   \item \cs{tl_replace_all:NvN} 不进入嵌套花括号组, |{ a { x & y } b }| 只得到
1451 %     |a{x&y}b|。这是够用的: 只有\emph{顶层}对齐符会破坏 \pkg{expl3} 条件式的括号组
1452 %     扫描 (实测组内含 catcode-4 |&| 时, 裸 \cs{tl_if_head_eq_meaning:nNTF}、
1453 %     \cs{tl_reverse:N}、\cs{tl_head:n}、\cs{tl_tail:n} 均不报错), 而
1454 %     \cs{__xeCJK_boundary_if_math_head:n} 递归进入组时会重新调用自身、再次经过本函数,
1455 %     下一层的 |&| 到那时才需要消解。
1456 %   \item \cs{__xeCJK_boundary_if_math_tail_space:n} 不走本函数, 因为它只做
1457 %     \cs{tl_trim_right_spaces:n} 与 \cs{tl_if_eq:NNTF}, 不做 head/tail 记号扫描,
1458 %     参数含 catcode-4 |&| 时实测 0 错误。需要消解的只有做扫描的
1459 %     \cs{__xeCJK_boundary_if_math_head:n} 与 \cs{__xeCJK_boundary_if_math_tail:n} 两个入口。
1460 % \end{itemize}
1461 %
1462 % \cs{tl_replace_all:Nnn} 及其变体 (这里用的是 \cs{tl_replace_all:NvN}) 按 catcode 与
1463 % charcode 双重相等匹配, 所以模式里的 |&|
1464 % 必须是 catcode 4 的对齐符。在本文件里直接写 |{ & }| 恰好可用, 但原因只是
1465 % \textbf{LaTeX} 环境默认把 |&| 设为 4 (\file{latex.ltx}:
1466 % |\catcode\&=4|), 而 \cs{ExplSyntaxOn} 不改 38 (实测它把 9、32 设为 9,
1467 % 58、95 设为 11, 126 设为 10, 而 38 前后都是 4)。注意
1468 % \cs{c_code_cctab} 之类的 catcode 表\emph{常量}与此无关: 它只在被
1469 % \cs{cctab_select:N} 选中时才生效, 本包从不选它。
1470 %
1471 % 关键在于\textbf{字面模式的类别}在本文件被读取 (tokenise) 时就冻结了, 与调用时的
1472 % \tn{catcode} 无关——加载后再改 |&| 的类别, 对字面写法和常量写法都没有影响。
1473 % 真正的风险是\emph{加载期}的 régime: 若 \tn{usepackage} 之前外层做过
1474 % |\catcode\&=12| (或在某个非默认 catcode 表下 \tn{input} 本包), 字面 |{ & }| 读进来
1475 % 就不是对齐符, 此后一律静默失配。 \cs{c__xeCJK_alignment_tl} 由
1476 % \cs{char_set_catcode_alignment:N} 在局部组里现场生成, 与读取时的 régime 无关,
1477 % 防住的正是这一种情形。
1478 \group_begin:
1479   \char_set_catcode_alignment:N \&
1480   \tl_const:Nn \c__xeCJK_alignment_tl { & }
1481 \group_end:
1482
1483 \cs_new_protected:Npn \__xeCJK_boundary_math_set:n #1
1484 {
1485   \tl_set:Nc \l__xeCJK_boundary_math_tl
1486     { \tl_trim_spaces:n {#1} }
1487   \tl_replace_all:NvN \l__xeCJK_boundary_math_tl
1488     \c__xeCJK_alignment_tl { \scan_stop: }
1489 }
1490
1491 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_head:n #1
1492 { T , F , TF }
1493 {
1494   \__xeCJK_boundary_math_set:n {#1}
1495   \tl_if_empty:NTF \l__xeCJK_boundary_math_tl
1496     { \prg_return_false: }
1497     {
1498       \exp_args:NV \tl_if_head_is_group:nTF
1499         \l__xeCJK_boundary_math_tl
1500         {
1501           \exp_args:Nc \__xeCJK_boundary_if_math_head:nTF
1502             { \tl_head:V \l__xeCJK_boundary_math_tl }
1503             { \prg_return_true: }
1504             { \prg_return_false: }

```



```

1505     }
1506     {
1507         \exp_args:NV \tl_if_head_eq_meaning:nNTF
1508         \l__xeCJK_boundary_math_tl $
1509         { \prg_return_true: }
1510     }
1511     \exp_args:NV \tl_if_head_eq_meaning:nNTF
1512     \l__xeCJK_boundary_math_tl \l
1513     { \prg_return_true: }
1514     {
1515         \exp_args:NV \tl_if_head_eq_meaning:nNTF
1516         \l__xeCJK_boundary_math_tl \ensuremath
1517         { \prg_return_true: }
1518         { \prg_return_false: }
1519     }
1520     }
1521     }
1522 }
1523 }
1524
1525 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_tail:n #1
1526 { T , F , TF }
1527 {
1528     \__xeCJK_boundary_math_set:n {#1}
1529     \tl_reverse:N \l__xeCJK_boundary_math_tl
1530     \exp_args:NV \__xeCJK_boundary_if_math_tail_reversed:nTF
1531     \l__xeCJK_boundary_math_tl
1532     { \prg_return_true: }
1533     { \prg_return_false: }
1534 }
1535
1536 % \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，
1537 % 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
1538 % 产生候选：未知宏也可能把末尾的分组、|$| 或 \tn{} 当作参数消费，
1539 % \cs{__xeCJK_boundary_math_end:n} 还要用当前列表的实际 math 节点或 marker
1540 % 确认，才会发布末类别。
1541 \prg_new_protected_conditional:Npnn
1542 \__xeCJK_boundary_if_math_tail_reversed:n #1 { T , F , TF }
1543 {
1544     \tl_if_empty:nTF {#1}
1545     { \prg_return_false: }
1546     {
1547         \tl_if_head_eq_meaning:nNTF {#1} $
1548         { \prg_return_true: }
1549         {
1550             \tl_if_head_eq_meaning:nNTF {#1} \
1551             { \prg_return_true: }
1552             {
1553                 \tl_if_head_is_group:nTF {#1}
1554                 {
1555                     \tl_set:Ne \l__xeCJK_boundary_math_tl
1556                     { \tl_trim_left_spaces:e { \tl_tail:n {#1} } }
1557                     \exp_args:NV \tl_if_head_eq_meaning:nNTF
1558                     \l__xeCJK_boundary_math_tl \ensuremath
1559                     { \prg_return_true: }
1560                     {
1561                         \exp_args:Ne \__xeCJK_boundary_if_math_tail:nTF
1562                         { \tl_head:n {#1} }
1563                         { \prg_return_true: }
1564                         { \prg_return_false: }
1565                     }
1566                 }
1567                 { \prg_return_false: }
1568             }
1569         }
1570     }
1571 }

```

```

1572
1573 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_edge:n #1
1574 { T , F , TF }
1575 {
1576   \__xeCJK_boundary_if_math_head:nTF {#1}
1577   { \prg_return_true: }
1578   {
1579     \__xeCJK_boundary_if_math_tail:nTF {#1}
1580     { \prg_return_true: }
1581     { \prg_return_false: }
1582   }
1583 }
1584
1585 \cs_new_protected:Npn \__xeCJK_boundary_math_begin:n #1
1586 {
1587   \__xeCJK_boundary_if_math_head:nT {#1}
1588   { \__xeCJK_boundary_math_begin_visible: }
1589 }
1590
1591 \cs_new_protected:Npn \__xeCJK_boundary_math_end:n #1
1592 {
1593   \__xeCJK_boundary_if_math_tail:nT {#1}
1594   {
1595     \__xeCJK_boundary_if_math_tail_space:nTF {#1}
1596     { \__xeCJK_boundary_math_end_confirm_space: }
1597     { \__xeCJK_boundary_math_end_confirm_node: }
1598   }
1599 }
1600
1601 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_tail_space:n #1
1602 { T , F , TF }
1603 {
1604   \tl_set:Nn \l__xeCJK_boundary_math_tl {#1}
1605   \tl_set:Nn \l__xeCJK_boundary_node_tl
1606   { \tl_trim_right_spaces:n {#1} }
1607   \tl_if_eq:NNTF \l__xeCJK_boundary_math_tl \l__xeCJK_boundary_node_tl
1608   { \prg_return_false: }
1609   { \prg_return_true: }
1610 }
1611
1612 \prg_new_conditional:Npnn \__xeCJK_boundary_if_last_math_node: { T , F , TF }
1613 {
1614   \__xeCJK_if_last_math:TF
1615   { \prg_return_true: }
1616   {
1617     \xeCJK_if_last_node:nTF { math }
1618     { \prg_return_true: }
1619     { \prg_return_false: }
1620   }
1621 }
1622
1623 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_node:
1624 {
1625   \__xeCJK_boundary_if_last_math_node:T
1626   { \__xeCJK_boundary_math_end_confirm: }
1627 }
1628
1629 % 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。
1630 % 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
1631 % 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
1632 % 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
1633 % 按普通 stream 或冻结空格分别处理。
1634 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_space:
1635 {
1636   \__xeCJK_if_last_glue:T
1637   {
1638     \skip_set_eq:NN \l__xeCJK_boundary_math_skip \tex_lastskip:D

```

```

1639     \__xeCJK_skip_if_interword:NT \l__xeCJK_boundary_math_skip
1640     {
1641         \tex_unskip:D
1642         \__xeCJK_boundary_if_last_math_node:T
1643         {
1644             \__xeCJK_boundary_math_end_confirm_space_visible:
1645         }
1646         \skip_horizontal:N \l__xeCJK_boundary_math_skip
1647     }
1648 }
1649 }
1650
1651 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_space_visible:
1652 {
1653     \__xeCJK_boundary_math_end_confirm:
1654     \tl_gset:cn
1655     {
1656         g__xeCJK_boundary_capture_
1657         \int_use:N \g__xeCJK_boundary_capture_depth_int
1658         _math_tail_space_flag_tl
1659     }
1660     { true }
1661     \tl_if_eq:cnT
1662     {
1663         g__xeCJK_boundary_capture_
1664         \int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl
1665     }
1666     { stream }
1667     {
1668         \tl_gset:ce
1669         {
1670             g__xeCJK_boundary_capture_
1671             \int_use:N \g__xeCJK_boundary_capture_depth_int
1672             _math_tail_space_skip_tl
1673         }
1674         { \skip_use:N \l__xeCJK_boundary_math_skip }
1675     }
1676 }
1677
1678 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm:
1679 {
1680     \int_compare:nNnT
1681     { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1682     {
1683         \int_compare:nNnT
1684         { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1685         {
1686             \tl_gset:cn
1687             {
1688                 g__xeCJK_boundary_capture_
1689                 \int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl
1690             }
1691             { math }
1692             \tl_gset:cn
1693             {
1694                 g__xeCJK_boundary_capture_
1695                 \int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
1696             }
1697             { true }
1698             \tl_gset:cn
1699             {
1700                 g__xeCJK_boundary_capture_
1701                 \int_use:N \g__xeCJK_boundary_capture_depth_int
1702                 _math_tail_space_flag_tl
1703             }
1704             { false }
1705         }
1706     }

```

```

1706     }
1707 }
1708
1709 \cs_new_protected:Npn \__xeCJK_boundary_capture_report_first:n #1
1710 {
1711     \int_compare:nNnT
1712     { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1713     {
1714         \int_compare:nNnT
1715         { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1716         {
1717             \int_step_inline:nn { \g__xeCJK_boundary_capture_depth_int }
1718             {
1719                 \tl_if_empty:cT { \g__xeCJK_boundary_capture_##1_first_tl }
1720                 {
1721                     \tl_gset:cn
1722                     { \g__xeCJK_boundary_capture_##1_first_tl } {#1}
1723                     \tl_if_eq:cnTF
1724                     { \g__xeCJK_boundary_capture_##1_kind_tl } { stream }
1725                     { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1726                     {
1727                         \tl_if_eq:cnT
1728                         { \g__xeCJK_boundary_capture_##1_kind_tl } { stream-ulem }
1729                         { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1730                     }
1731                 }
1732             }
1733         }
1734     }
1735 }
1736
1737 \cs_new_protected:Npn \__xeCJK_boundary_capture_emit_left:nn #1#2
1738 {
1739     \tl_set:Ne \l__xeCJK_boundary_node_tl
1740     { \tl_use:c { \g__xeCJK_boundary_capture_#1_before_tl } }
1741     \__xeCJK_boundary_emit_left:nVn
1742     {#1} \l__xeCJK_boundary_node_tl {#2}
1743     \tl_gset:cn
1744     { \g__xeCJK_boundary_capture_#1_space_flag_tl } { false }
1745 }
1746
1747 \cs_new_protected:Npn \__xeCJK_boundary_capture_suspend:
1748 {
1749     \int_gincr:N \g__xeCJK_boundary_suspend_depth_int
1750     \tl_if_exist:cF
1751     { \g__xeCJK_boundary_suspend_int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1752     {
1753         \tl_new:c
1754         { \g__xeCJK_boundary_suspend_int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1755         \tl_new:c
1756         { \g__xeCJK_boundary_suspend_int_use:N \g__xeCJK_boundary_suspend_depth_int _pending_tl }
1757     }
1758     \exp_args:Nc \tl_gset_eq:NN
1759     { \g__xeCJK_boundary_suspend_int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1760     \g__xeCJK_last_node_tl
1761     \tl_gset:ce
1762     { \g__xeCJK_boundary_suspend_int_use:N \g__xeCJK_boundary_suspend_depth_int _pending_tl }
1763     { \bool_if:NTF \g__xeCJK_glue_check_pending_bool { true } { false } }
1764 }
1765
1766 \cs_new_protected:Npn \__xeCJK_boundary_capture_resume:
1767 {
1768     \tl_gset:Ne \g__xeCJK_last_node_tl
1769     {
1770         \tl_use:c
1771         { \g__xeCJK_boundary_suspend_int_use:N \g__xeCJK_boundary_suspend_depth_int _node_tl }
1772     }

```

```

1773 \tl_if_eq:cnTF
1774 { g__xeCJK_boundary_suspend\int_use:N \g__xeCJK_boundary_suspend_depth_int _pending_tl }
1775 { true }
1776 { \bool_gset_true:N \g__xeCJK_glue_check_pending_bool }
1777 { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
1778 \int_gdecr:N \g__xeCJK_boundary_suspend_depth_int
1779 }
1780
1781 \cs_new_protected:Npn \__xeCJK_boundary_inline_box_begin:
1782 {
1783   \int_compare:nNnTF
1784   { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1785   { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1786   {
1787     \mode_if_math:TF
1788     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1789     {
1790       \mode_leave_vertical:
1791       \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1792       \__xeCJK_boundary_box_begin:
1793     }
1794   }
1795 }
1796
1797 \cs_new_protected:Npn \__xeCJK_boundary_inline_box_end:n #1
1798 {
1799   \seq_gpop:NN
1800   \g__xeCJK_boundary_active_seq
1801   \l__xeCJK_boundary_active_tl
1802   \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1803   { \__xeCJK_boundary_box_end:n {#1} }
1804 }
1805
1806 \cs_new_protected:Npn \__xeCJK_boundary_inline_last_box_begin:
1807 {
1808   \int_compare:nNnTF
1809   { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1810   { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1811   {
1812     \mode_if_math:TF
1813     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1814     {
1815       \mode_leave_vertical:
1816       \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1817       \__xeCJK_boundary_capture_begin:
1818       \tl_gset:cn
1819       { g__xeCJK_boundary_capture\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
1820       { box }
1821     }
1822   }
1823 }
1824
1825 \cs_new_protected:Npn \__xeCJK_boundary_inline_last_box_end:n #1
1826 {
1827   \seq_gpop:NN
1828   \g__xeCJK_boundary_active_seq
1829   \l__xeCJK_boundary_active_tl
1830   \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1831   { \__xeCJK_boundary_last_box_end:n {#1} }
1832 }
1833
1834 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_begin:n #1
1835 { \__xeCJK_boundary_inline_stream_begin:nn {#1} { stream } }
1836
1837 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_begin:nn #1#2
1838 {
1839   \int_compare:nNnTF

```

```

1840 { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1841 { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1842 {
1843   \mode_if_math:TF
1844   { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1845   {
1846     \mode_leave_vertical:
1847     \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1848     \__xeCJK_boundary_capture_begin:
1849     \tl_gset:cn
1850     { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
1851     {#2}
1852     \str_case:nn {#1}
1853     {
1854       { default } { \__xeCJK_boundary_capture_class:n { default } }
1855       { first-default } { \__xeCJK_boundary_capture_class:n { default } }
1856     }
1857   }
1858 }
1859 }
1860
1861 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_end:n #1
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     \scan_stop:
1869     \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1870     \tl_set:Ne \l__xeCJK_boundary_first_tl
1871     {
1872       \tl_use:c
1873       { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _first_tl }
1874     }
1875     \tl_set:Ne \l__xeCJK_boundary_last_tl
1876     {
1877       \tl_use:c
1878       { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl }
1879     }
1880     % 流式命令若在最外层留下 xeCJK marker，就以该 marker 记录的
1881     % 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
1882     % marker 移到外层列表的命令；没有末尾 marker 的 annotation/math
1883     % 仍使用 capture 观察值。
1884     \exp_args:NV
1885     \__xeCJK_boundary_set_last_from_node:n \l__xeCJK_boundary_node_tl
1886     \__xeCJK_boundary_set_math_tail_from_capture:
1887     % |default| 的开始 hook 已经固定首类别；这里仅覆写末类别即可让
1888     % stream 两端都为 Default，同时不丢弃其他模式实际观察到的首类别。
1889     \str_if_eq:nnT {#1} { default }
1890     { \tl_set:Nn \l__xeCJK_boundary_last_tl {#1} }
1891     \tl_gclear:N \g__xeCJK_last_node_tl
1892     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1893     \tl_if_empty:NTF \l__xeCJK_boundary_first_tl
1894     {
1895       \__xeCJK_boundary_replay_before:
1896     }
1897     {
1898       \tl_if_empty:NT \l__xeCJK_boundary_last_tl
1899       { \tl_set_eq:NN \l__xeCJK_boundary_last_tl \l__xeCJK_boundary_first_tl }
1900       \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_last_tl
1901     }
1902     \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
1903   }
1904 }
1905
1906 \cs_new_protected:Npn \__xeCJK_boundary_hmode_transparent_begin:

```

```

1907 {
1908   \int_compare:nNnTF
1909     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1910     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1911     {
1912       \mode_if_horizontal:TF
1913       {
1914         \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1915         \__xeCJK_boundary_capture_begin:
1916         \tl_gset:cn
1917         { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
1918         { transparent }
1919       }
1920       { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1921     }
1922   }
1923
1924 \cs_new_protected:Npn \__xeCJK_boundary_hmode_transparent_end:
1925 {
1926   \seq_gpop:NN
1927   \g__xeCJK_boundary_active_seq
1928   \l__xeCJK_boundary_active_tl
1929   \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1930   {
1931     \__xeCJK_boundary_replay_before:
1932     \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
1933   }
1934 }
1935
1936 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent:
1937 {
1938   \mode_if_horizontal:T
1939   {
1940     \tl_if_empty:NF \g__xeCJK_last_node_tl
1941     {
1942       \__xeCJK_if_last_hlist:TF
1943       {
1944         \box_set_to_last:N \l__xeCJK_boundary_post_box
1945         \dim_compare:nNnTF
1946         {
1947           \box_wd:N \l__xeCJK_boundary_post_box
1948           + \box_ht:N \l__xeCJK_boundary_post_box
1949           + \box_dp:N \l__xeCJK_boundary_post_box
1950         }
1951         = { \c_zero_dim }
1952         { \__xeCJK_boundary_post_transparent_relocate: }
1953         { \box_use_drop:N \l__xeCJK_boundary_post_box }
1954       }
1955       {
1956         \xeCJK_if_last_node:TF
1957         {
1958           \exp_args:NV
1959           \__xeCJK_boundary_replay_node:n \g__xeCJK_last_node_tl
1960         }
1961         { }
1962       }
1963     }
1964   }
1965 }
1966
1967 % 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
1968 % 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
1969 % 末尾 glue，再按节点证据检查其下方的 marker。命中后把“marker + glue”整体
1970 % 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
1971 % 时按原顺序还原，不能把任意 glue 搬过 hbox。普通 stream 的  $\text{space}$ 
1972 % 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
1973 % 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。

```

```

1974 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent_relocate:
1975 {
1976   \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1977   \tl_if_empty:NTF \l__xeCJK_boundary_node_tl
1978   {
1979     \__xeCJK_if_last_glue:TF
1980     { \__xeCJK_boundary_post_transparent_relocate_glue: }
1981     { \box_use_drop:N \l__xeCJK_boundary_post_box }
1982   }
1983   {
1984     % 盒子直接盖住 marker，中间没有 glue。
1985     \box_use_drop:N \l__xeCJK_boundary_post_box
1986     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
1987     { \__xeCJK_boundary_math_space_expire: }
1988     {
1989       \exp_args:NV
1990       \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
1991     }
1992   }
1993 }
1994
1995 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent_relocate_glue:
1996 {
1997   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1998   \tex_unskip:D
1999   \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2000   \tl_if_empty:NTF \l__xeCJK_boundary_node_tl
2001   {
2002     \skip_horizontal:N \l__xeCJK_last_skip
2003     \box_use_drop:N \l__xeCJK_boundary_post_box
2004   }
2005   {
2006     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
2007     {
2008       % |math-space| 过期时不再移动后缀；先恢复候选 glue，再放回
2009       % 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。
2010       \skip_horizontal:N \l__xeCJK_last_skip
2011       \box_use_drop:N \l__xeCJK_boundary_post_box
2012       \__xeCJK_boundary_math_space_expire:
2013     }
2014     {
2015       \box_use_drop:N \l__xeCJK_boundary_post_box
2016       \exp_args:NV
2017       \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
2018       \skip_horizontal:N \l__xeCJK_last_skip
2019     }
2020   }
2021 }
2022
2023 \cs_new_protected:Npn \__xeCJK_boundary_capture_begin:
2024 {
2025   \int_gincr:N \g__xeCJK_boundary_capture_depth_int
2026   \int_compare:nNnT
2027   { \g__xeCJK_boundary_capture_depth_int } > { 2 }
2028   {
2029     \exp_args:Ne \__xeCJK_boundary_capture_allocate:n
2030     { \int_use:N \g__xeCJK_boundary_capture_depth_int }
2031   }
2032   \tl_gclear:c
2033   { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl }
2034   \tl_gclear:c
2035   {
2036     \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int
2037     _before_math_space_skip_tl
2038   }
2039   \tl_gclear:c
2040   { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _first_tl }

```



```

2041 \tl_gclear:c
2042 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl }
2043 \tl_gclear:c
2044 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
2045 % 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
2046 % 已求值的 skip 数值序列化到本层 tl, 不展开原始 \CJKglue/\CJKecglue 定义。
2047 \xeCJK_glue_to_skip:nN
2048 { \tex_kern:D \c_zero_dim \xeCJK_space_glue: }
2049 \l__xeCJK_tmp_skip
2050 \tl_gset:ce
2051 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_skip_tl }
2052 { \skip_use:N \l__xeCJK_tmp_skip }
2053 \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_tmp_skip
2054 \tl_gset:ce
2055 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _ecglue_skip_tl }
2056 { \skip_use:N \l__xeCJK_tmp_skip }
2057 \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_tmp_skip
2058 \tl_gset:ce
2059 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _ccglue_skip_tl }
2060 { \skip_use:N \l__xeCJK_tmp_skip }
2061 \tl_gset:cn
2062 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_tl }
2063 { false }
2064 \tl_gset:ce
2065 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _reserve_space_f }
2066 { \bool_if:NTF \l__xeCJK_reserve_space_bool { true } { false } }
2067 \tl_gset:ce
2068 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _xecglue_flag_tl }
2069 { \bool_if:NTF \l__xeCJK_xecglue_bool { true } { false } }
2070 \tl_gset:cn
2071 { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag }
2072 { false }
2073 \tl_gset:cn
2074 {
2075     g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int
2076     _math_tail_space_flag_tl
2077 }
2078 { false }
2079 \tl_gclear:c
2080 {
2081     g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int
2082     _math_tail_space_skip_tl
2083 }
2084 \__xeCJK_boundary_capture_space:
2085 \scan_stop:
2086 \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2087 \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { math-space }
2088 {
2089     \__xeCJK_if_last_glue:TF
2090     {
2091         \tl_gset:ce
2092         {
2093             g__xeCJK_boundary_capture_
2094             \int_use:N \g__xeCJK_boundary_capture_depth_int
2095             _before_math_space_skip_tl
2096         }
2097         { \skip_use:N \tex_lastskip:D }
2098     }
2099     {
2100         \tl_gset:ce
2101         {
2102             g__xeCJK_boundary_capture_
2103             \int_use:N \g__xeCJK_boundary_capture_depth_int
2104             _before_math_space_skip_tl
2105         }
2106         { \skip_use:N \l__xeCJK_boundary_math_actual_skip }
2107     }

```

```

2108     }
2109     \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { CJK-space }
2110     {
2111         \tl_gset:cn
2112         { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2113           { true }
2114         }
2115     \tl_if_empty:NT \l__xeCJK_boundary_node_tl
2116     {
2117         \tl_if_eq:cnT
2118         { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2119           { true }
2120         {
2121             \__xeCJK_boundary_restore_space:
2122             \tl_gset:cn
2123             { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2124               { false }
2125           }
2126         }
2127     \exp_args:Nc \tl_gset_eq:NN
2128     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl }
2129     \l__xeCJK_boundary_node_tl
2130     \tl_gclear:N \g__xeCJK_last_node_tl
2131     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2132 }
2133
2134 \cs_new_protected:Npn \__xeCJK_boundary_box_begin:
2135 {
2136     \__xeCJK_boundary_capture_begin:
2137     \tl_gset:cn
2138     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl }
2139     { box }
2140     \exp_args:Nc \__xeCJK_boundary_box_open:N
2141     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2142 }
2143
2144 \cs_new_protected:Npn \__xeCJK_boundary_box_open:N #1
2145 {
2146     \tex_global:D \tex_setbox:D #1
2147     \tex_hbox:D \c_group_begin_token
2148 }
2149
2150 \cs_new_protected:Npn \__xeCJK_boundary_capture_space:
2151 {
2152     \__xeCJK_if_last_glue:T
2153     {
2154         \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
2155         \__xeCJK_skip_if_interword:NT \l__xeCJK_last_skip
2156         {
2157             \tl_gset:ce
2158             { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_skip_tl
2159               { \skip_use:N \l__xeCJK_last_skip }
2160             \tl_gset:cn
2161             { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2162               { true }
2163             \tex_unskip:D
2164           }
2165         }
2166     }
2167
2168 \cs_new_protected:Npn \__xeCJK_boundary_pop_node:N #1
2169 {
2170     \tl_clear:N #1
2171     \xeCJK_if_last_node:TF
2172     {
2173         \dim_case:nnF { \l__xeCJK_last_kern_dim }
2174         {

```

```

2175 { \_xeCJK\_node:n { CJK } } { \tl\_set:Nn #1 { CJK } }
2176 { \_xeCJK\_node:n { CJK-space } } { \tl\_set:Nn #1 { CJK-space } }
2177 { \_xeCJK\_node:n { CJK-widow } } { \tl\_set:Nn #1 { CJK-widow } }
2178 { \_xeCJK\_node:n { default } } { \tl\_set:Nn #1 { default } }
2179 { \_xeCJK\_node:n { math } } { \tl\_set:Nn #1 { math } }
2180 { \_xeCJK\_node:n { math-space } }
2181 {
2182   \tl\_set:Nn #1 { math-space }
2183   \_xeCJK\_boundary\_math\_space\_decode:
2184 }
2185 { \_xeCJK\_node:n { math-space-frozen } }
2186 { \tl\_set:Nn #1 { math-space-frozen } }
2187 { \_xeCJK\_node:n { normalspace } } { \tl\_set:Nn #1 { normalspace } }
2188 }
2189 { \_xeCJK\_make\_node:N \l\_xeCJK\_last\_kern\_dim }
2190 }
2191 {
2192   \xeCJK\_if\_last\_node:nT { default-space }
2193   {
2194     \xeCJK\_remove\_node:
2195     \tl\_set:Nn #1 { default-space }
2196   }
2197 }
2198 }
2199
2200 \prg\_new\_conditional:Npnn \_xeCJK\_boundary\_if\_last\_math\_space: { T , F , TF }
2201 {
2202   \xeCJK\_if\_last\_node:nTF { math-space }
2203   { \prg\_return\_true: }
2204   {
2205     \xeCJK\_if\_last\_node:nTF { math-space-frozen }
2206     { \prg\_return\_true: }
2207     { \prg\_return\_false: }
2208   }
2209 }
2210
2211 \cs\_new\_protected:Npn \_xeCJK\_boundary\_math\_space\_consume:
2212 {
2213   \xeCJK\_if\_last\_node:nTF { math-space }
2214   {
2215     \xeCJK\_remove\_node:
2216     \_xeCJK\_boundary\_math\_space\_decode:
2217   }
2218   { \xeCJK\_remove\_node: }
2219   \bool\_gset\_false:N \g\_xeCJK\_glue\_check\_pending\_bool
2220 }
2221
2222 % |math-space| 后面的两对零净宽 kern 依次保存普通 stream 空格的伸长量
2223 % 和收缩量。marker 位于最外侧，移除 marker 后按相反次序取回两对 kern。
2224 % 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。
2225 \cs\_new\_protected:Npn \_xeCJK\_boundary\_math\_space\_set\_actual:n #1
2226 {
2227   \skip\_set:Nn \l\_xeCJK\_boundary\_math\_actual\_skip {#1}
2228   \dim\_set:Nn \l\_xeCJK\_boundary\_math\_stretch\_dim
2229   { \tex\_gluestretch:D \l\_xeCJK\_boundary\_math\_actual\_skip }
2230   \dim\_set:Nn \l\_xeCJK\_boundary\_math\_shrink\_dim
2231   { \tex\_glueshrink:D \l\_xeCJK\_boundary\_math\_actual\_skip }
2232   \skip\_set:Nn \l\_xeCJK\_boundary\_math\_actual\_skip
2233   {
2234     \c\_zero\_dim
2235     plus \l\_xeCJK\_boundary\_math\_stretch\_dim
2236     minus \l\_xeCJK\_boundary\_math\_shrink\_dim
2237   }
2238 }
2239
2240 \cs\_new\_protected:Npn \_xeCJK\_boundary\_math\_space\_encode:
2241 {

```

```

2242 \dim_set:Nn \l__xeCJK_boundary_math_stretch_dim
2243 { \tex_gluestretch:D \l__xeCJK_boundary_math_actual_skip }
2244 \dim_set:Nn \l__xeCJK_boundary_math_shrink_dim
2245 { \tex_glueshrink:D \l__xeCJK_boundary_math_actual_skip }
2246 \__xeCJK_make_node:N \l__xeCJK_boundary_math_stretch_dim
2247 \__xeCJK_make_node:N \l__xeCJK_boundary_math_shrink_dim
2248 }
2249
2250 \cs_new_protected:Npn \__xeCJK_boundary_math_space_decode:
2251 {
2252   \dim_zero:N \l__xeCJK_boundary_math_stretch_dim
2253   \dim_zero:N \l__xeCJK_boundary_math_shrink_dim
2254   \xeCJK_if_last_node:TF
2255   {
2256     \dim_set_eq:NN
2257     \l__xeCJK_boundary_math_shrink_dim \l__xeCJK_last_kern_dim
2258     \xeCJK_if_last_node:TF
2259     {
2260       \dim_set_eq:NN
2261       \l__xeCJK_boundary_math_stretch_dim \l__xeCJK_last_kern_dim
2262     }
2263     { }
2264   }
2265   { }
2266   \skip_set:Nn \l__xeCJK_boundary_math_actual_skip
2267   {
2268     \c_zero_dim
2269     plus \l__xeCJK_boundary_math_stretch_dim
2270     minus \l__xeCJK_boundary_math_shrink_dim
2271   }
2272 }
2273
2274 % 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
2275 % 计算，伸缩差额则扣除参数内实际空格的伸缩量。box、wrapped-box 和
2276 % 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，
2277 % 外层只能扣除词间空格的自然宽度，并完整保留 |CJKe glue| 自己的伸缩分量。
2278 % 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
2279 % 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
2280 % 自身的固有宽度。
2281 \cs_new_protected:Npn \__xeCJK_boundary_math_space_set_delta:n #1
2282 {
2283   \str_if_eq:nnTF {#1} { math-space-frozen }
2284   {
2285     \dim_set:Nn \l__xeCJK_boundary_math_space_dim
2286     { \l__xeCJK_boundary_math_skip }
2287     \dim_set:Nn \l__xeCJK_boundary_math_ecglue_dim
2288     { \l__xeCJK_boundary_math_ecglue_skip }
2289     \dim_set:Nn \l__xeCJK_tmp_dim
2290     {
2291       \dim_max:nn
2292       {
2293         \l__xeCJK_boundary_math_ecglue_dim
2294         - \l__xeCJK_boundary_math_space_dim
2295       }
2296       { \c_zero_dim }
2297     }
2298     \skip_set:Nn \l__xeCJK_boundary_math_delta_skip
2299     {
2300       \l__xeCJK_boundary_math_ecglue_skip
2301       - \l__xeCJK_boundary_math_ecglue_dim
2302       + \l__xeCJK_tmp_dim
2303     }
2304   }
2305   {
2306     \dim_set:Nn \l__xeCJK_boundary_math_space_dim
2307     { \l__xeCJK_boundary_math_skip }
2308     \dim_set:Nn \l__xeCJK_boundary_math_ecglue_dim

```

```

2309         { \l__xeCJK_boundary_math_ecglue_skip }
2310     \dim_set:Nn \l__xeCJK_tmp_dim
2311     {
2312         \l__xeCJK_boundary_math_ecglue_dim
2313         - \l__xeCJK_boundary_math_space_dim
2314     }
2315     \skip_set:Nn \l__xeCJK_boundary_math_delta_skip
2316     {
2317         \l__xeCJK_boundary_math_ecglue_skip
2318         - \l__xeCJK_boundary_math_ecglue_dim
2319         + \l__xeCJK_tmp_dim
2320         - \l__xeCJK_boundary_math_actual_skip
2321     }
2322 }
2323 }
2324
2325 \cs_new_protected:Npn \__xeCJK_boundary_math_space_to_CJK:
2326 {
2327     \xeCJK_if_last_node:nTF { math-space-frozen }
2328     { \tl_set:Nn \l__xeCJK_boundary_math_type_tl { math-space-frozen } }
2329     { \tl_set:Nn \l__xeCJK_boundary_math_type_tl { math-space } }
2330     \__xeCJK_boundary_math_space_consume:
2331     \bool_if:NT \l__xeCJK_xecglue_bool
2332     {
2333         \skip_set:Nn \l__xeCJK_boundary_math_skip
2334         { \c_xeCJK_space_skip_tl }
2335         \skip_set_eq:NN
2336         \l__xeCJK_boundary_math_ecglue_skip \l__xeCJK_ecglue_skip
2337         \exp_args:NV \__xeCJK_boundary_math_space_set_delta:n
2338         \l__xeCJK_boundary_math_type_tl
2339         \skip_if_eq:nnF
2340         { \l__xeCJK_boundary_math_delta_skip } { \c_zero_skip }
2341         { \skip_horizontal:N \l__xeCJK_boundary_math_delta_skip }
2342     }
2343 }
2344
2345 \cs_new_protected:Npn \__xeCJK_boundary_box_end:n #1
2346 {
2347     \c_group_end_token
2348     \__xeCJK_boundary_box_finish:nn {#1} { wrapped }
2349 }
2350
2351 \cs_new_protected:Npn \__xeCJK_boundary_last_box_end:n #1
2352 {
2353     \__xeCJK_if_last_hlist:TF
2354     {
2355         \exp_args:Nc \box_set_to_last:N
2356         { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2357         \__xeCJK_boundary_box_finish:nn {#1} { last }
2358     }
2359     {
2360         \tl_gclear:N \g__xeCJK_last_node_tl
2361         \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2362         \__xeCJK_boundary_replay_before:
2363         \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
2364     }
2365 }
2366
2367 \int_new:N \g__xeCJK_boundary_probe_type_int
2368 \box_new:N \l__xeCJK_boundary_probe_box
2369 \tl_new:N \l__xeCJK_boundary_probe_name_tl
2370 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_capture_box_sized:
2371 { T , F , TF }
2372 {
2373     \tl_set:Ne \l__xeCJK_boundary_probe_name_tl
2374     {
2375         g__xeCJK_boundary_capture_

```

```

2376     \int_use:N \g__xeCJK_boundary_capture_depth_int _box
2377   }
2378   \bool_lazy_and:nnTF
2379   {
2380     ! \dim_compare_p:nNn
2381       { \box_wd:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2382   }
2383   {
2384     \bool_lazy_or_p:nn
2385     {
2386       ! \dim_compare_p:nNn
2387         { \box_ht:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2388     }
2389     {
2390       ! \dim_compare_p:nNn
2391         { \box_dp:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2392     }
2393   }
2394   { \prg_return_true: }
2395   { \prg_return_false: }
2396 }
2397 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_capture_box_visible:
2398 { T , F , TF }
2399 {
2400   \__xeCJK_boundary_if_capture_box_sized:TF
2401   {
2402     % 只复制节点，不重扫 token，无排版副作用。
2403     \hbox_set:Nn \l__xeCJK_boundary_probe_box
2404     {
2405       \hbox_unpack:c { \l__xeCJK_boundary_probe_name_tl }
2406       \int_gset:Nn \g__xeCJK_boundary_probe_type_int
2407         { \tex_lastnodetype:D }
2408     }
2409     \box_clear:N \l__xeCJK_boundary_probe_box
2410     % \lastnodetype 编码沿用 eTeX 手册的节点类型表 (0 传统 char、
2411     % 1 hlist、3 rule、9 whatsit、10 math、11 glue、12 kern) ;
2412     % XeTeX 把 native 文本实现为 word 节点、同样报 9，但它必然已被
2413     % interchar 观察，不会进入这里。这里把 char、rule、math，以及
2414     % 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。
2415     \int_case:nnTF { \g__xeCJK_boundary_probe_type_int }
2416     {
2417       { 0 } { }
2418       { 3 } { }
2419       { 10 } { }
2420       { 12 } { }
2421     }
2422     { \prg_return_true: }
2423     { \prg_return_false: }
2424   }
2425   { \prg_return_false: }
2426 }
2427
2428 \tl_new:N \g__xeCJK_boundary_probe_last_tl
2429 \cs_new_protected:Npn \__xeCJK_boundary_box_set_last_from_node:
2430 {
2431   \tl_set:Ne \l__xeCJK_boundary_probe_name_tl
2432   {
2433     g__xeCJK_boundary_capture_
2434     \int_use:N \g__xeCJK_boundary_capture_depth_int _box
2435   }
2436   \tl_gclear:N \g__xeCJK_boundary_probe_last_tl
2437   \hbox_set:Nn \l__xeCJK_boundary_probe_box
2438   {
2439     \hbox_unpack:c { \l__xeCJK_boundary_probe_name_tl }
2440     \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2441     \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { math-space }
2442     { \tl_set:Nn \l__xeCJK_boundary_node_tl { math-space-frozen } }

```

```

2443     \tl_gset_eq:NN
2444     \g__xeCJK_boundary_probe_last_tl \l__xeCJK_boundary_node_tl
2445   }
2446   \box_clear:N \l__xeCJK_boundary_probe_box
2447   \exp_args:NV
2448     \__xeCJK_boundary_set_last_from_node:n \g__xeCJK_boundary_probe_last_tl
2449 }
2450
2451 \cs_new_protected:Npn \__xeCJK_boundary_box_finish:nn #1#2
2452 {
2453   \tl_set:Nc \l__xeCJK_boundary_first_tl
2454   {
2455     \tl_use:c
2456     { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _first_tl }
2457   }
2458   \tl_set:Nc \l__xeCJK_boundary_last_tl
2459   {
2460     \tl_use:c
2461     { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl }
2462   }
2463   \tl_gclear:N \g__xeCJK_last_node_tl
2464   \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2465   % 嵌套 box 结束时写入的 marker 会留在当前盒子的节点列表末尾。
2466   % 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
2467   % 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
2468   % 推断直接写进所有外层 capture。
2469   % 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
2470   % 使观察值不准，此时应以节点列表末尾的 marker 为准。
2471   \__xeCJK_boundary_box_set_last_from_node:
2472   \__xeCJK_boundary_set_math_tail_from_capture:
2473   % 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
2474   % 的首尾类别；stream 路径则由开始 hook 固定首类别、结束 hook 固定
2475   % 末类别，两条路径的公开语义相同。
2476   \str_if_eq:nnT {#1} { default }
2477   {
2478     \tl_set:Nn \l__xeCJK_boundary_first_tl { default }
2479     \tl_set:Nn \l__xeCJK_boundary_last_tl { default }
2480   }
2481   % rule 和尚未由公式观察器记录的 math 不触发 interchar 转换。若盒子
2482   % 直接排出了可见内容
2483   % （宽度非零、高度或深度非零，且末节点为 char/rule/math/kern），
2484   % 却没有观察到任何类别，就按 Default 首尾重建边界。空 makebox、strut
2485   % 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
2486   % 输出处理（issue 998）。
2487   % 只把 default 报告给尚未取得首类别的外层 capture，使 \nfss@text
2488   % 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
2489   % 节点列表末尾由本层写入的 marker，并据此更新末类别。
2490   \tl_if_empty:NT \l__xeCJK_boundary_first_tl
2491   {
2492     \__xeCJK_boundary_if_capture_box_visible:T
2493     {
2494       \tl_set:Nn \l__xeCJK_boundary_first_tl { default }
2495       \tl_set:Nn \l__xeCJK_boundary_last_tl { default }
2496       \__xeCJK_boundary_capture_report_first:n { default }
2497     }
2498   }
2499   \tl_if_empty:NTF \l__xeCJK_boundary_first_tl
2500   { \__xeCJK_boundary_box_end_transparent:n {#2} }
2501   {
2502     \tl_set:Nc \l__xeCJK_boundary_node_tl
2503     {
2504       \tl_use:c
2505       { \g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl }
2506     }
2507     \exp_args:NVV \__xeCJK_boundary_emit_left:nn
2508       \l__xeCJK_boundary_node_tl
2509       \l__xeCJK_boundary_first_tl

```

```

2510     \box_use_drop:c
2511     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2512     \tl_if_empty:NT \l__xeCJK_boundary_last_tl
2513     { \tl_set_eq:NN \l__xeCJK_boundary_last_tl \l__xeCJK_boundary_first_tl }
2514     \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_last_tl
2515   }
2516   \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
2517 }
2518
2519 \cs_new_protected:Npn \__xeCJK_boundary_box_end_transparent:n #1
2520 {
2521   \str_if_eq:nnTF {#1} { wrapped }
2522   {
2523     \hbox_unpack_drop:c
2524     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2525   }
2526   {
2527     \box_use_drop:c
2528     { g__xeCJK_boundary_capture_\int_use:N \g__xeCJK_boundary_capture_depth_int _box }
2529   }
2530   \__xeCJK_boundary_replay_before:
2531 }
2532
2533 \cs_new_protected:Npn \__xeCJK_boundary_replay_before:
2534 {
2535   \tl_set:Ne \l__xeCJK_boundary_node_tl
2536   {
2537     \tl_use:c
2538     {
2539       g__xeCJK_boundary_capture_
2540       \int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl
2541     }
2542   }
2543   \clist_if_in:nVTF
2544   { math-space , math-space-frozen } \l__xeCJK_boundary_node_tl
2545   {
2546     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
2547     { \__xeCJK_boundary_replay_before_math_space: }
2548     { \__xeCJK_boundary_replay_node:n { math-space-frozen } }
2549   }
2550   {
2551     \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
2552     \__xeCJK_boundary_restore_space:
2553   }
2554 }
2555
2556 % 普通 stream 的 |math-space| 只在真实参数空格仍紧邻列表末尾时重放。
2557 % transparent 命令若已经在二者之间写入 special、零尺寸盒子等节点，直接
2558 % 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，
2559 % 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
2560 % 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。
2561 \cs_new_protected:Npn \__xeCJK_boundary_replay_before_math_space:
2562 {
2563   \__xeCJK_if_last_glue:T
2564   {
2565     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
2566     \skip_if_eq:nnT
2567     { \l__xeCJK_last_skip }
2568     {
2569       \tl_use:c
2570       {
2571         g__xeCJK_boundary_capture_
2572         \int_use:N \g__xeCJK_boundary_capture_depth_int
2573         _before_math_space_skip_tl
2574       }
2575     }
2576   }

```



```

2577         \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2578         {
2579             \tl_use:c
2580             {
2581                 g__xeCJK_boundary_capture_
2582                 \int_use:N \g__xeCJK_boundary_capture_depth_int
2583                 _before_math_space_skip_tl
2584             }
2585         }
2586         \__xeCJK_boundary_replay_node:n { math-space }
2587     }
2588 }
2589 }
2590
2591 \cs_new_protected:Npn \__xeCJK_boundary_math_space_expire:
2592 {
2593     \tl_gclear:N \g__xeCJK_last_node_tl
2594     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2595 }
2596
2597 \cs_new_protected:Npn \__xeCJK_boundary_emit_left:nn #1#2
2598 {
2599     \__xeCJK_boundary_emit_left:nnn
2600     { \int_use:N \g__xeCJK_boundary_capture_depth_int } {#1} {#2}
2601 }
2602
2603 \cs_new_protected:Npn \__xeCJK_boundary_emit_left:nnn #1#2#3
2604 {
2605     \clist_if_in:nnTF { math-space , math-space-frozen } {#2}
2606     { \__xeCJK_boundary_emit_left_math_space:nnn {#1} {#2} {#3} }
2607     { \__xeCJK_boundary_emit_left_normal:nnn {#1} {#2} {#3} }
2608 }
2609 \cs_new_protected:Npn \__xeCJK_boundary_emit_left_math_space:nnn #1#2#3
2610 {
2611     \str_if_eq:nnT {#3} { CJK }
2612     {
2613         \tl_if_eq:cnT
2614         { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2615         {
2616             \str_if_eq:nnT {#2} { math-space }
2617             {
2618                 \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2619                 {
2620                     \tl_use:c
2621                     { g__xeCJK_boundary_capture_#1_before_math_space_skip_tl }
2622                 }
2623             }
2624             \skip_set:Nn \l__xeCJK_boundary_math_skip
2625             { \tl_use:c { g__xeCJK_boundary_capture_#1_space_skip_tl } }
2626             \skip_set:Nn \l__xeCJK_boundary_math_ecglue_skip
2627             { \tl_use:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl } }
2628             \__xeCJK_boundary_math_space_set_delta:n {#2}
2629             \skip_if_eq:nnF
2630             { \l__xeCJK_boundary_math_delta_skip } { \c_zero_skip }
2631             {
2632                 \__xeCJK_boundary_use_glue:nn {#1}
2633                 { \l__xeCJK_boundary_math_delta_skip }
2634             }
2635         }
2636     }
2637 }
2638 \cs_new_protected:Npn \__xeCJK_boundary_emit_left_normal:nnn #1#2#3
2639 {
2640     \clist_if_in:nnTF { CJK , CJK-space , CJK-widow } {#2}
2641     {
2642         \str_if_eq:nnTF {#3} { math }
2643         {

```

```

2644 \tl_if_eq:cnTF
2645 { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2646 {
2647   \tl_if_eq:cnTF
2648   { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2649   { \__xeCJK_boundary_use_ecglue:n {#1} }
2650   { \__xeCJK_boundary_restore_space:n {#1} }
2651 }
2652 { \__xeCJK_boundary_use_ecglue:n {#1} }
2653 }
2654 {
2655   \str_if_eq:nnTF {#3} { CJK }
2656   {
2657     \str_if_eq:nnT {#2} { CJK-widow }
2658     { \xeCJK_widow_penalty: }
2659     \tl_if_eq:cnTF
2660     { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2661     {
2662       \tl_if_eq:cnTF
2663       { g__xeCJK_boundary_capture_#1_reserve_space_flag_tl } { true }
2664       { \__xeCJK_boundary_restore_space:n {#1} }
2665       { \__xeCJK_boundary_use_ccglue:n {#1} }
2666     }
2667     { \__xeCJK_boundary_use_ccglue:n {#1} }
2668   }
2669   { \__xeCJK_boundary_use_ecglue:n {#1} }
2670 }
2671 }
2672 {
2673   \clist_if_in:nnTF { default , default-space , math } {#2}
2674   {
2675     \str_if_eq:nnTF {#3} { CJK }
2676     {
2677       \tl_if_eq:cnTF
2678       { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2679       {
2680         \tl_if_eq:cnTF
2681         { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2682         { \__xeCJK_boundary_use_ecglue:n {#1} }
2683         { \__xeCJK_boundary_restore_space:n {#1} }
2684       }
2685       { \__xeCJK_boundary_use_ecglue:n {#1} }
2686     }
2687     { \__xeCJK_boundary_restore_space:n {#1} }
2688   }
2689   { \__xeCJK_boundary_restore_space:n {#1} }
2690 }
2691 }
2692 \cs_generate_variant:Nn \__xeCJK_boundary_emit_left:nnn { nVn }
2693
2694 \cs_new_protected:Npn \__xeCJK_boundary_use_ecglue:n #1
2695 {
2696   \__xeCJK_boundary_use_glue:nn {#1}
2697   { \tl_use:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl } }
2698 }
2699
2700 \cs_new_protected:Npn \__xeCJK_boundary_use_ccglue:n #1
2701 {
2702   \__xeCJK_boundary_use_glue:nn {#1}
2703   { \tl_use:c { g__xeCJK_boundary_capture_#1_ccglue_skip_tl } }
2704 }
2705
2706 \cs_new_protected:Npn \__xeCJK_boundary_use_glue:nn #1#2
2707 {
2708   \tl_if_eq:cnTF
2709   { g__xeCJK_boundary_capture_#1_kind_tl } { stream-ulem }
2710   { \use:c { __xeCJK_boundary_use_ulem_glue:n } {#2} }

```

```

2711     { \skip_horizontal:n {#2} }
2712   }
2713
2714 \cs_new_protected:Npn \__xeCJK_boundary_restore_space:
2715 {
2716   \__xeCJK_boundary_restore_space:n
2717   { \int_use:N \g__xeCJK_boundary_capture_depth_int }
2718 }
2719
2720 \cs_new_protected:Npn \__xeCJK_boundary_restore_space:n #1
2721 {
2722   \tl_if_eq:cnT
2723   { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2724   {
2725     \__xeCJK_boundary_use_glue:nn {#1}
2726     { \tl_use:c { g__xeCJK_boundary_capture_#1_space_skip_tl } }
2727   }
2728 }
2729
2730 \cs_new_protected:Npn \__xeCJK_boundary_set_last_from_node:n #1
2731 {
2732   \clist_if_in:nnTF { CJK , CJK-space , CJK-widow } {#1}
2733   { \tl_set:Nn \l__xeCJK_boundary_last_tl { CJK } }
2734   {
2735     \clist_if_in:nnT
2736     { default , default-space } {#1}
2737     { \tl_set:Nn \l__xeCJK_boundary_last_tl { default } }
2738     \str_if_eq:nnT {#1} { math }
2739     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math } }
2740     \str_if_eq:nnT {#1} { math-space }
2741     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space } }
2742     \str_if_eq:nnT {#1} { math-space-frozen }
2743     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space-frozen } }
2744   }
2745 }
2746
2747 \cs_new_protected:Npn \__xeCJK_boundary_set_math_tail_from_capture:
2748 {
2749   \tl_if_eq:cnT
2750   {
2751     g__xeCJK_boundary_capture_
2752     \int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
2753   }
2754   { true }
2755   {
2756     \tl_if_eq:cnTF
2757     {
2758       g__xeCJK_boundary_capture_
2759       \int_use:N \g__xeCJK_boundary_capture_depth_int
2760       _math_tail_space_flag_tl
2761     }
2762     { true }
2763     {
2764       \tl_if_eq:cnTF
2765       {
2766         g__xeCJK_boundary_capture_
2767         \int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl
2768       }
2769       { stream }
2770       {
2771         \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space }
2772         \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2773         {
2774           \tl_use:c
2775           {
2776             g__xeCJK_boundary_capture_
2777             \int_use:N \g__xeCJK_boundary_capture_depth_int

```

```

2778         _math_tail_space_skip_tl
2779     }
2780 }
2781 }
2782 { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space-frozen } }
2783 }
2784 { \tl_set:Nn \l__xeCJK_boundary_last_tl { math } }
2785 }
2786 }
2787
2788 \cs_new_protected:Npn \__xeCJK_boundary_replay_node:n #1
2789 {
2790     % Default 字符在盒子内部更新的 spacefactor 不会传播到外层列表; stream
2791     % 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
2792     % 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
2793     % 错用于盒外以 1000 生成的源码空格。
2794     \clist_if_in:nnT
2795     {
2796         default , default-space , math , math-space , math-space-frozen ,
2797         normalspace
2798     }
2799     {#1}
2800     { \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D }
2801     \str_if_eq:nnTF {#1} { math-space }
2802     {
2803         \__xeCJK_boundary_math_space_encode:
2804         \xeCJK_make_node:n {#1}
2805         \__xeCJK_boundary_enable_source_space:n {#1}
2806     }
2807     {
2808         \str_if_eq:nnTF {#1} { math-space-frozen }
2809         {
2810             \xeCJK_make_node:n {#1}
2811             \__xeCJK_boundary_enable_source_space:n {#1}
2812         }
2813         {
2814             \str_if_eq:nnTF {#1} { default-space }
2815             {
2816                 \__xeCJK_make_space_node:
2817                 \tl_gset:Nn \g__xeCJK_last_node_tl { default }
2818             }
2819             {
2820                 \clist_if_in:nnT
2821                 { CJK , CJK-space , CJK-widow , default , math , normalspace }
2822                 {#1}
2823                 {
2824                     \xeCJK_make_node:n {#1}
2825                     \__xeCJK_boundary_enable_source_space:n {#1}
2826                 }
2827             }
2828         }
2829     }
2830 }
2831 \cs_new_protected:Npn \__xeCJK_boundary_enable_source_space:n #1
2832 {
2833     \clist_if_in:nnT
2834     {
2835         CJK , CJK-space , CJK-widow , math , math-space ,
2836         math-space-frozen
2837     }
2838     {#1}
2839     { \bool_gset_true:N \g__xeCJK_glue_check_pending_bool }
2840 }
2841
2842 \cs_new_protected:Npn \__xeCJK_boundary_register_box:nn #1#2
2843 {
2844     \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}

```

```

2845     {
2846       \prop_gput:Nnn \g__xeCJK_boundary_registered_prop {#1} {#2}
2847       \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2848       { \__xeCJK_boundary_inline_last_box_begin: }
2849       \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2850       { \__xeCJK_boundary_inline_last_box_end:n {#2} }
2851     }
2852   }
2853
2854 \cs_new_protected:Npn \__xeCJK_boundary_register_wrapped_box:nn #1#2
2855 {
2856   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2857   {
2858     \prop_gput:Nnn
2859     \g__xeCJK_boundary_registered_prop {#1} { wrapped-box/#2 }
2860     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2861     { \__xeCJK_boundary_inline_box_begin: }
2862     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2863     { \__xeCJK_boundary_inline_box_end:n {#2} }
2864   }
2865 }
2866
2867 \cs_new_protected:Npn \__xeCJK_boundary_register_transparent:n #1
2868 {
2869   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2870   {
2871     \prop_gput:Nnn
2872     \g__xeCJK_boundary_registered_prop {#1} { transparent }
2873     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2874     { \__xeCJK_boundary_hmode_transparent_begin: }
2875     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2876     { \__xeCJK_boundary_hmode_transparent_end: }
2877   }
2878 }
2879
2880 \cs_new_protected:Npn \__xeCJK_boundary_register_post_transparent:n #1
2881 {
2882   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2883   {
2884     \prop_gput:Nnn
2885     \g__xeCJK_boundary_registered_prop {#1} { post-transparent }
2886     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2887     { \__xeCJK_boundary_post_transparent: }
2888   }
2889 }
2890
2891 \cs_new_protected:Npn \__xeCJK_boundary_register_stream:nn #1#2
2892 {
2893   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2894   {
2895     \prop_gput:Nnn \g__xeCJK_boundary_registered_prop {#1} { stream/#2 }
2896     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2897     { \__xeCJK_boundary_inline_stream_begin:n {#2} }
2898     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2899     { \__xeCJK_boundary_inline_stream_end:n {#2} }
2900   }
2901 }
2902
2903 % 通用注册函数把命令写入 \cs{g__xeCJK_boundary_registered_prop}。另有一些
2904 % 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
2905 % 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
2906 % 已有专用适配器外层。
2907 \cs_new_protected:Npn \__xeCJK_boundary_reserve:n #1
2908 { \prop_gput:Nnn \g__xeCJK_boundary_reserved_prop {#1} { adapter } }
2909 \cs_new_protected:Npn \__xeCJK_boundary_reserve:N #1
2910 {
2911   \exp_args:Ne \__xeCJK_boundary_reserve:n { \cs_to_str:N #1 }

```

```

2912 }
2913 \prg_new_conditional:Npnn \__xeCJK_boundary_if_registered:n #1 { T , F , TF }
2914 {
2915   \prop_if_in:NnTF \g__xeCJK_boundary_registered_prop {#1}
2916   { \prg_return_true: }
2917   {
2918     \prop_if_in:NnTF \g__xeCJK_boundary_reserved_prop {#1}
2919     { \prg_return_true: }
2920     { \prg_return_false: }
2921   }
2922 }
2923
2924 \__xeCJK_msg_new:nn { boundary-register-command }
2925 {
2926   The~value~of~`command'~in~`experiment/boundary-register'~must~be~
2927   exactly~one~control~sequence.
2928 }
2929 \__xeCJK_msg_new:nn { boundary-register-choice }
2930 { Invalid~boundary~registration~choice:~`#1. }
2931 \__xeCJK_msg_new:nn { boundary-register-missing }
2932 { The~`#1'~entry~is~required~in~`experiment/boundary-register'. }
2933 \__xeCJK_msg_new:nn { boundary-register-mode }
2934 { The~boundary~registration~strategy~and~mode~cannot~be~combined:~`#1. }
2935 \__xeCJK_msg_new:nn { boundary-register-duplicate }
2936 { The~command~`#1'~has~already~been~declared~for~boundary~registration. }
2937 \__xeCJK_msg_new:nn { boundary-register-conflict }
2938 { The~command~`#1'~already~has~an~xeCJK~boundary~registration. }
2939 \__xeCJK_msg_new:nn { boundary-register-late }
2940 {
2941   The~`experiment/boundary-register'~option~can~only~be~used~before~
2942   the~end~of~the~preamble.
2943 }
2944 \__xeCJK_msg_new:nn { boundary-register-undefined }
2945 { The~registered~boundary~command~`#1'~is~still~undefined. }
2946
2947 \cs_new_protected:Npn \__xeCJK_boundary_user_register:n #1
2948 {
2949   \bool_if:NTF \g__xeCJK_boundary_user_register_closed_bool
2950   { \__xeCJK_error:n { boundary-register-late } }
2951   {
2952     \bool_set_false:N \l__xeCJK_boundary_user_invalid_bool
2953     \bool_set_false:N \l__xeCJK_boundary_user_mode_given_bool
2954     \tl_clear:N \l__xeCJK_boundary_user_name_tl
2955     \tl_clear:N \l__xeCJK_boundary_user_strategy_tl
2956     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { auto }
2957     \keys_set:nn { xeCJK / boundary-register } {#1}
2958     \tl_if_empty:NT \l__xeCJK_boundary_user_name_tl
2959     {
2960       \__xeCJK_error:ne { boundary-register-missing } { command }
2961       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2962     }
2963     \tl_if_empty:NT \l__xeCJK_boundary_user_strategy_tl
2964     {
2965       \__xeCJK_error:ne { boundary-register-missing } { strategy }
2966       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2967     }
2968     \bool_if:NF \l__xeCJK_boundary_user_invalid_bool
2969     { \__xeCJK_boundary_user_register_validate: }
2970     \bool_if:NF \l__xeCJK_boundary_user_invalid_bool
2971     { \__xeCJK_boundary_user_register_store: }
2972   }
2973 }
2974
2975 \cs_new_protected:Npn \__xeCJK_boundary_user_register_command:n #1
2976 {
2977   \tl_if_single_token:NTF {#1}
2978   {

```

```

2979     \token_if_cs:NTF #1
2980     { \tl_set:Nx \l__xeCJK_boundary_user_name_tl { \cs_to_str:N #1 } }
2981     {
2982         \__xeCJK_error:n { boundary-register-command }
2983         \tl_set:Nn \l__xeCJK_boundary_user_name_tl { invalid }
2984         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2985     }
2986 }
2987 {
2988     \__xeCJK_error:n { boundary-register-command }
2989     \tl_set:Nn \l__xeCJK_boundary_user_name_tl { invalid }
2990     \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2991 }
2992 }
2993
2994 \cs_new_protected:Npn \__xeCJK_boundary_user_register_strategy:n #1
2995 {
2996     \str_case:nnF {#1}
2997     {
2998         { box } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2999         { wrapped-box } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
3000         { stream } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
3001         { transparent } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
3002         { post-transparent } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
3003     }
3004     {
3005         \__xeCJK_error:ne { boundary-register-choice }
3006         { strategy~=~\tl_to_str:n {#1} }
3007         \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl { invalid }
3008         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3009     }
3010 }
3011
3012 \cs_new_protected:Npn \__xeCJK_boundary_user_register_mode:n #1
3013 {
3014     \bool_set_true:N \l__xeCJK_boundary_user_mode_given_bool
3015     \str_case:nnF {#1}
3016     {
3017         { auto } { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
3018         { default } { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
3019         { first-default } { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
3020     }
3021     {
3022         \__xeCJK_error:ne { boundary-register-choice }
3023         { mode~=~\tl_to_str:n {#1} }
3024         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3025     }
3026 }
3027
3028 \cs_new_protected:Npn \__xeCJK_boundary_user_register_validate:
3029 {
3030     \str_case:Vn \l__xeCJK_boundary_user_strategy_tl
3031     {
3032         { box }
3033         {
3034             \tl_if_eq:NnT \l__xeCJK_boundary_user_mode_tl { first-default }
3035             {
3036                 \__xeCJK_error:ne { boundary-register-mode } { box/first-default }
3037                 \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3038             }
3039         }
3040         { wrapped-box }
3041         {
3042             \tl_if_eq:NnT \l__xeCJK_boundary_user_mode_tl { first-default }
3043             {
3044                 \__xeCJK_error:ne { boundary-register-mode }
3045                 { wrapped-box/first-default }

```

```

3046         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3047     }
3048 }
3049 { stream } { }
3050 { transparent }
3051 {
3052     \bool_if:NT \l__xeCJK_boundary_user_mode_given_bool
3053     {
3054         \__xeCJK_error:ne { boundary-register-mode }
3055         { transparent/\l__xeCJK_boundary_user_mode_tl }
3056         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3057     }
3058     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { none }
3059 }
3060 { post-transparent }
3061 {
3062     \bool_if:NT \l__xeCJK_boundary_user_mode_given_bool
3063     {
3064         \__xeCJK_error:ne { boundary-register-mode }
3065         { post-transparent/\l__xeCJK_boundary_user_mode_tl }
3066         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3067     }
3068     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { none }
3069 }
3070 }
3071 }
3072
3073 \cs_new_protected:Npn \__xeCJK_boundary_user_register_store:
3074 {
3075     \exp_args:NV \__xeCJK_boundary_if_registered:nTF
3076     \l__xeCJK_boundary_user_name_tl
3077     {
3078         \__xeCJK_error:ne { boundary-register-conflict }
3079         { \c_backslash_str \l__xeCJK_boundary_user_name_tl }
3080     }
3081     {
3082         \exp_args:NNV \prop_if_in:NnTF
3083         \g__xeCJK_boundary_user_registered_prop \l__xeCJK_boundary_user_name_tl
3084         {
3085             \__xeCJK_error:ne { boundary-register-duplicate }
3086             { \c_backslash_str \l__xeCJK_boundary_user_name_tl }
3087         }
3088         {
3089             \tl_set:Ne \l__xeCJK_boundary_user_value_tl
3090             {
3091                 \l__xeCJK_boundary_user_strategy_tl
3092                 / \l__xeCJK_boundary_user_mode_tl
3093             }
3094             \exp_args:NNVV \prop_gput:Nnn
3095             \g__xeCJK_boundary_user_registered_prop
3096             \l__xeCJK_boundary_user_name_tl
3097             \l__xeCJK_boundary_user_value_tl
3098         }
3099     }
3100 }
3101
3102 \cs_new_protected:Npn \__xeCJK_boundary_user_register_apply:
3103 {
3104     \bool_gset_true:N \g__xeCJK_boundary_user_register_closed_bool
3105     \prop_map_inline:Nn \g__xeCJK_boundary_user_registered_prop
3106     {
3107         \__xeCJK_boundary_if_registered:nTF {##1}
3108         {
3109             \__xeCJK_error:ne { boundary-register-conflict }
3110             { \c_backslash_str ##1 }
3111         }
3112         {

```



```

3113         \__xeCJK_boundary_user_register_apply:nn {##1} {##2}
3114         \seq_gput_right:Nn \g__xeCJK_boundary_user_applied_seq {##1}
3115     }
3116 }
3117 }
3118
3119 \cs_new_protected:Npn \__xeCJK_boundary_user_register_apply:nn #1#2
3120 {
3121     \seq_set_split:Nnn \l__xeCJK_boundary_user_value_seq { / } {#2}
3122     \seq_pop_left:NN
3123         \l__xeCJK_boundary_user_value_seq \l__xeCJK_boundary_user_strategy_tl
3124     \seq_pop_left:NN
3125         \l__xeCJK_boundary_user_value_seq \l__xeCJK_boundary_user_mode_tl
3126     \str_case:Vn \l__xeCJK_boundary_user_strategy_tl
3127     {
3128         { box }
3129         {
3130             \exp_args:NnV \__xeCJK_boundary_register_box:nn
3131                 {#1} \l__xeCJK_boundary_user_mode_tl
3132         }
3133         { wrapped-box }
3134         {
3135             \exp_args:NnV \__xeCJK_boundary_register_wrapped_box:nn
3136                 {#1} \l__xeCJK_boundary_user_mode_tl
3137         }
3138         { stream }
3139         {
3140             \exp_args:NnV \__xeCJK_boundary_register_stream:nn
3141                 {#1} \l__xeCJK_boundary_user_mode_tl
3142         }
3143         { transparent }
3144         { \__xeCJK_boundary_register_transparent:n {#1} }
3145         { post-transparent }
3146         { \__xeCJK_boundary_register_post_transparent:n {#1} }
3147     }
3148 }
3149
3150 \cs_new_protected:Npn \__xeCJK_boundary_user_register_check:
3151 {
3152     \seq_map_inline:Nn \g__xeCJK_boundary_user_applied_seq
3153     {
3154         \cs_if_exist:cF {##1}
3155         {
3156             \__xeCJK_error:ne { boundary-register-undefined }
3157             { \c_backslash_str ##1 }
3158         }
3159     }
3160 }
3161
3162 % \tn{mbox} 和 \tn{fbox} 的公开命令是 robust wrapper; 实际单参数正文由带
3163 % 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
3164 % 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
3165 % 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
3166 % \tn{setbox} 测量。
3167 \cs_new_protected:Npn \__xeCJK_boundary_prepare_onearg_box:N #1
3168 {
3169     \group_begin: \exp_args:NcNc \group_end:
3170     { \__xeCJK_boundary_prepare_onearg_box:NN }
3171     #1 { \cs_to_str:N #1 ~ }
3172 }
3173 \cs_new_protected:Npn \__xeCJK_boundary_prepare_onearg_box:NN #1#2
3174 {
3175     \__xeCJK_boundary_reserve:N #2
3176     \exp_args:Nc \cs_new_eq:NN
3177     { \__xeCJK_boundary_orig_\cs_to_str:N #1 _box:n } #2
3178     \cs_gset_protected:Npn #2 ##1
3179     {

```

```

3180     \_xeCJK_boundary_math_begin:n {##1}
3181     \use:c { \_xeCJK_boundary_orig_\cs_to_str:N #1 _box:n }
3182     { ##1 \_xeCJK_boundary_math_end:n {##1} }
3183   }
3184 }
3185
3186 % 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
3187 % transparent capture 的同一 begin/end，只负责保留零参数或单参数调用签名。
3188 \cs_new_protected:Npn \_xeCJK_boundary_wrap_transparent_noarg:NN #1#2
3189 {
3190   \cs_if_exist:NF #1
3191   {
3192     \_xeCJK_boundary_reserve:N #2
3193     \cs_set_eq:NN #1#2
3194     \cs_gset_protected:Npn #2
3195     {
3196       \_xeCJK_boundary_hmode_transparent_begin:
3197       #1
3198       \_xeCJK_boundary_hmode_transparent_end:
3199     }
3200   }
3201 }
3202
3203 \cs_new_protected:Npn \_xeCJK_boundary_wrap_transparent_onearg:NN #1#2
3204 {
3205   \cs_if_exist:NF #1
3206   {
3207     \_xeCJK_boundary_reserve:N #2
3208     \cs_set_eq:NN #1#2
3209     \cs_gset_protected:Npn #2 ##1
3210     {
3211       \_xeCJK_boundary_hmode_transparent_begin:
3212       #1 ##1
3213       \_xeCJK_boundary_hmode_transparent_end:
3214     }
3215   }
3216 }

```

上面那个包装以 #1 ##1 的形式转发参数，适用于原函数只是顺序执行参数内容的情形。若原函数会把参数再次用作 \hbox:n 一类的参数，丢失花括号就会改变分组：参数后面相邻的记号会被吸进参数内部。hyperref 的 _hyp_target_raise:n 属于后者，它把参数交给 \hbox:n，无花括号转发会让紧随其后的 \Hy@SaveSpaceFactor 被卷进 \hyper@anchorstart 的参数，结果把 \spacefactor 赋值写进 pdf:dest 名字、并把锚点名排成可见文本。

这与 xeCJK 的钩子无关：不挂任何钩子、仅做无花括号透传也能复现（实测盒宽 81.16002pt 对 oracle 41.66002pt，该读数来自「不挂钩子的纯透传」对照；在包装变体里误用无花括号版则为 84.49002pt）。因此只需要一个保留花括号的转发变体。

```

3217 \cs_new_protected:Npn \_xeCJK_boundary_wrap_transparent_onearg_braced:NN #1#2
3218 {
3219   \cs_if_exist:NF #1
3220   {
3221     \_xeCJK_boundary_reserve:N #2
3222     \cs_set_eq:NN #1#2
3223     \cs_gset_protected:Npn #2 ##1
3224     {
3225       \_xeCJK_boundary_hmode_transparent_begin:
3226       #1 {##1}
3227       \_xeCJK_boundary_hmode_transparent_end:
3228     }
3229   }
3230 }
3231
3232 % \changes{v3.10.4}{2026/07/20}{包装内核私有 \tn{@imakebox} 与
3233 % \tn{@iframebox} 的 |[\#1][\#2]\#3| 签名以注册 \tn{makebox} /
3234 % \tn{framebox}；内核若调整该签名须同步此 wrapper (\#992)。}

```

```

3235 % 这里包装的是本包最低支持内核 \LaTeX{} 2026-06-01 中, 内部
3236 % \tn{@imakebox}/\tn{@iframbox} 的 |[#1][#2]#3| 参数签名, 不是稳定公共
3237 % 接口。若内核改变该签名, 必须同步 wrapper; \texttt{command-boundary01}
3238 % 的 optional \tn{makebox}/\tn{framebox} 场景作为漂移门禁。
3239 \cs_new_protected:Npn \__xeCJK_boundary_register_makeboxes:
3240 {
3241   \clist_map_inline:nn { makebox , framebox , @imakebox , @iframbox }
3242   { \__xeCJK_boundary_reserve:n {##1} }
3243   \cs_if_exist:NF \__xeCJK_orig_imakebox:w
3244   {
3245     \cs_new_eq:NN \__xeCJK_orig_imakebox:w \@imakebox
3246     \cs_gset_protected:Npn \@imakebox [##1][##2]##3
3247     {
3248       \__xeCJK_boundary_inline_last_box_begin:
3249       \__xeCJK_boundary_math_begin:n {##3}
3250       \__xeCJK_orig_imakebox:w [##1][##2]
3251       { ##3 \__xeCJK_boundary_math_end:n {##3} }
3252       \__xeCJK_boundary_inline_last_box_end:n { auto }
3253     }
3254   }
3255   \cs_if_exist:NF \__xeCJK_orig_iframbox:w
3256   {
3257     \cs_new_eq:NN \__xeCJK_orig_iframbox:w \@iframbox
3258     \cs_gset_protected:Npn \@iframbox [##1][##2]##3
3259     {
3260       \__xeCJK_boundary_inline_last_box_begin:
3261       \__xeCJK_boundary_math_begin:n {##3}
3262       \__xeCJK_orig_iframbox:w [##1][##2]
3263       { ##3 \__xeCJK_boundary_math_end:n {##3} }
3264       \__xeCJK_boundary_inline_last_box_end:n { auto }
3265     }
3266   }
3267 }
3268
3269 % \changes{v3.10.5}{2026/07/29}{\tn{sbox} 改用专用适配器: \tn{global}
3270 % 等前缀必须紧邻 \tn{setbox}, 放进 \texttt{cmd/sbox/before} 钩子的赋值
3271 % 会把前缀吃掉, 使 \tn{global}\tn{sbox} 静默退化为局部赋值 (\#1029)。}
3272 % \tn{sbox} 不能使用通用 \texttt{cmd/.../before} 钩子。钩子代码插在
3273 % \tn{setbox} 之前执行, 而调用方写的 \tn{global} 是等待下一个赋值的前缀:
3274 % 钩子里任何赋值都会把它消耗掉, 于是 \tn{global}\tn{sbox} 退化为局部赋值,
3275 % 盒子在分组结束时丢失, 且不报任何错误 (\#1029, \pkg{algorithm2e} 的
3276 % \tn{caption} 即如此消失)。因此这里直接包装内部入口, 把暂停观察移到盒子
3277 % 内部, 使前缀始终紧邻 \tn{setbox}。
3278 %
3279 % \tn{savebox} 的四种形式 (无可选参数、|[wd]|、|[wd][pos]|, 以及 picture 形式
3280 % |(x,y)[pos]|, 后者经 \tn{@isavepicbox} 汇入) 最终都走同一个 \cs{sbox~} 入口,
3281 % 因此排版路径一并
3282 % 覆盖; 但 \tn{global}\tn{savebox} 跨分组本来就不生效 (前缀在 \tn{savebox}
3283 % 自己的 \tn{@ifnextchar} 前瞻阶段即被消耗, 原版 \LaTeX{} 亦然), 不在本次
3284 % 修复范围内。
3285 %
3286 % 与 \cs{color@bex}、\tn{@textcolor} 等适配器不同, 这里不保存并委托原定义, 而是
3287 % 整体覆盖 \cs{sbox~}: 内核 \cs{sbox~} 本体就是赋值语句, 委托写法必须把
3288 % \tn{setbox} 留在最外层才能保住前缀, 代价明显高于收益。代价是若某个包在本包
3289 % \emph{之前} 也包装了 \cs{sbox~}, 它的包装会被丢弃 (反序加载正常)。撰写本节时
3290 % 扫描过完整 \TeX{} Live 树, 未发现任何宏包重定义 \cs{sbox~} 或对 \tn{sbox}
3291 % 安装 \texttt{cmd/sbox} 钩子, 故接受这一取舍; \cs{cs_if_eq:cNF} 保证本包自身
3292 % 不会重复包装。
3293 \cs_new_protected:Npn \__xeCJK_boundary_sbox:Nn #1#2
3294 {
3295   \tex_setbox:D #1 \tex_hbox:D
3296   {
3297     \__xeCJK_boundary_capture_suspend:
3298     \color@setgroup #2 \color@endgroup
3299     \__xeCJK_boundary_capture_resume:
3300   }
3301 }

```

```

3302 \cs_new_protected:Npn \__xeCJK_boundary_prepare_sbox:
3303 {
3304   \cs_if_eq:cNF { sbox~ } \__xeCJK_boundary_sbox:Nn
3305   { \cs_gset_eq:cN { sbox~ } \__xeCJK_boundary_sbox:Nn }
3306 }
3307
3308 \__xeCJK_boundary_prepare_onearg_box:N \mbox
3309 \__xeCJK_boundary_prepare_onearg_box:N \fbox
3310 \__xeCJK_boundary_register_box:nn { mbox } { auto }
3311 \__xeCJK_boundary_register_box:nn { fbox } { auto }
3312 \__xeCJK_boundary_register_makeboxes:
3313 \__xeCJK_boundary_register_post_transparent:n { null }
3314 \__xeCJK_boundary_reserve:n { sbox }
3315 \__xeCJK_boundary_prepare_sbox:

```

\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。

```

3316 \cs_new_eq:NN \xeCJK_make_space_node: \prg_do_nothing:
3317 \cs_new_protected:Npe \__xeCJK_make_space_node:
3318 {
3319   \tex_hskip:D - \__xeCJK_glue_node:n { default-space }
3320   \tex_hskip:D \__xeCJK_glue_node:n { default-space }
3321 }

```

CJKglue CJK 文字之间插入的 glue。

```

3322 \keys_define:nn { xeCJK / options }
3323 {
3324   CJKglue .code:n =
3325   {
3326     \cs_set_protected:Npn \CJKglue {#1}
3327     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
3328   }
3329 }
3330 \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

```

3331 \keys_define:nn { xeCJK / options }
3332 {
3333   CJKecglue .code:n =
3334   {
3335     \cs_set_protected:Npn \CJKecglue {#1}
3336     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
3337   } ,
3338   xCJKecglue .choice: ,
3339   xCJKecglue / true .code:n =
3340   {
3341     \bool_set_true:N \l__xeCJK_xecglue_bool
3342     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
3343     \cs_set_eq:NN \xeCJK_make_space_node: \__xeCJK_make_space_node:
3344     \cs_set_eq:NN \xeCJK_check_for_xglue: \__xeCJK_check_for_xglue:
3345     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_xecglue:
3346     \cs_set_eq:NN
3347     \xeCJK_check_for_ecglue_normalsp:
3348     \__xeCJK_check_for_xecglue_normalsp:
3349   } ,
3350   xCJKecglue / false .code:n =
3351   {
3352     \bool_set_false:N \l__xeCJK_xecglue_bool
3353     \cs_set_eq:NN \xeCJK_space_or_xecglue: \xeCJK_space_glue:
3354     \xeCJK_cs_clear:N \xeCJK_make_space_node:
3355     \xeCJK_cs_clear:N \xeCJK_check_for_xglue:
3356     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:
3357     \cs_set_eq:NN

```

```

3358     \xeCJK_check_for_ecglue_normalsp:
3359     \_xeCJK_check_for_ecglue_normalsp:
3360   } ,
3361   xCJKecglue / unknown .code:n =
3362   {
3363     \bool_set_true:N \l__xeCJK_xecglue_bool
3364     \cs_set_protected:Npn \CJKecglue {#1}
3365     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
3366     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
3367     \cs_set_eq:NN \xeCJK_make_space_node: \_xeCJK_make_space_node:
3368     \cs_set_eq:NN \xeCJK_check_for_xglue: \_xeCJK_check_for_xglue:
3369     \cs_set_eq:NN \xeCJK_check_for_ecglue: \_xeCJK_check_for_xecglue:
3370     \cs_set_eq:NN
3371     \xeCJK_check_for_ecglue_normalsp:
3372     \_xeCJK_check_for_xecglue_normalsp:
3373   } ,
3374   xCJKecglue .default:n = { true }
3375 }
3376 \cs_new_eq:NN \xeCJK_space_glue: \c_space_tl
3377 \skip_new:N \l__xeCJK_ecglue_skip
3378 \bool_new:N \l__xeCJK_xecglue_bool

```

CJKspace 是否保留 CJK 文字间的空白, 默认不保留。

```

3379 \keys_define:nn { xeCJK / options }
3380 {
3381   CJKspace .choice: ,
3382   CJKspace / true .code:n =
3383   {
3384     \bool_set_true:N \l__xeCJK_reserve_space_bool
3385     \cs_set_protected:Npn \_xeCJK_ccglue_or_space:
3386     { \xeCJK_space_glue: }
3387   } ,
3388   CJKspace / false .code:n =
3389   {
3390     \bool_set_false:N \l__xeCJK_reserve_space_bool
3391     \cs_set_protected:Npn \_xeCJK_ccglue_or_space:
3392     { \CJKglue }
3393   } ,
3394   CJKspace .default:n = { true } ,
3395   space .meta:n = { CJKspace = true } ,
3396   nospace .meta:n = { CJKspace = false }
3397 }
3398 \bool_new:N \l__xeCJK_reserve_space_bool
3399 \xeCJK_inter_class_toks:nnn { CJK } { Boundary } { \xeCJK_CJK_and_Boundary:w }

```

\xeCJK_CJK_and_Boundary:w 当边界是 \relax 的时候, 它可能是由 \csname ... \endcsname 的形式产生的, 这样就可能出现问题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。

```

3400 \cs_new_protected:Npn \xeCJK_CJK_and_Boundary:w
3401 {
3402   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
3403   { \_xeCJK_CJK_and_math: }
3404   {
3405     \group_align_safe_begin:
3406     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3407     {
3408       \token_if_macro:NTF \l_peek_token
3409       {
3410         \token_if_eq_meaning:NNTF
3411         \l_peek_token \_xeCJK_boundary_identity:n
3412         { \_xeCJK_boundary_peek_math:w }
3413         { \_xeCJK_boundary_reserve_space: }

```

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

```

3414     }
3415     {
3416         \token_if_group_begin:NTF \l_peek_token
3417         { \__xeCJK_boundary_group_math:w }
3418         { \__xeCJK_boundary_group_end:n { CJK-space } }
3419     }
3420 }
3421 {
3422     \token_if_eq_meaning:NNTF \l_peek_token \scan_stop:
3423     { \__xeCJK_CJK_and_Boundary_relax:N }
3424     {
3425         \token_if_group_begin:NTF \l_peek_token
3426         { \__xeCJK_boundary_group_math:w }
3427         {
3428             \token_if_group_end:NTF \l_peek_token
3429             {
3430                 \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
3431                 \__xeCJK_boundary_group_end:n { CJK }
3432             }
3433             {
3434                 \token_if_macro:NTF \l_peek_token
3435                 {
3436                     \token_if_eq_meaning:NNTF
3437                     \l_peek_token \__xeCJK_boundary_identity:n
3438                     { \__xeCJK_boundary_peek_math:w }
3439                     { \__xeCJK_boundary_group_end:n { CJK } }
3440                 }
3441                 { \__xeCJK_boundary_group_end:n { CJK } }
3442             }
3443         }
3444     }
3445 }
3446 }
3447 }
3448 \cs_new_protected:Npn \__xeCJK_CJK_and_math:
3449 {
3450     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3451     { \xeCJK_class_group_end: \xeCJK_space_or_xecglue: }
3452     { \xeCJK_class_group_end: \CJKecglue }
3453 }
3454 \cs_new_protected:Npn \__xeCJK_boundary_CJK_and_math:
3455 { \group_align_safe_end: \__xeCJK_CJK_and_math: }

```

这里只需要知道左花括号之后的第一个记号是不是 `\$`, 因此不能用 `n` 型参数把整个分组吞掉: 分组的内容可能是别的宏正在执行的语法, 提前吞掉再重新发出会破坏它。

具体的事故是 #1038: XeTeX 判断字符类时会展开宏, 包括 `\protected` 宏——`\protected` 只对 `\edef` 一类的完全展开语境有效, 对「不断展开直到取得一个不可展开记号」的类别前瞻无效。于是 `tabular` 里中文 `\` 的 `\`

(已 `\let` 为 `@tabularcr`) 被展开, 其替换文本 `{\ifnum0=`}\fi...` 的首个记号是显式左花括号, 本函数因此被选中, 并把 `{\ifnum0=`}` 整个吞掉。重新发出时用的是隐式的 `\c_group_end_token`, 而 `latex` 那个平衡花括号的技巧要求反引号紧跟显式字符记号}, 于是报 `Improper alphabetic constant`。

改为用 `\afterassignment` 加 `\let` 把那一枚左花括号当作赋值的右值吸收——这不开新分组——再前瞻其后的第一个记号, 最后补发一枚隐式左花括号恢复分组。除这一枚花括号之外, 输入流原样保留, 因此 ``}` 的相邻关系不被破坏。

```

3456 \cs_new_protected:Npn \__xeCJK_boundary_group_math:w
3457 {
3458     \tex_afterassignment:D \__xeCJK_boundary_group_math_peek:
3459     \tex_let:D \__xeCJK_boundary_group_math_brace:
3460 }
3461 \cs_new_eq:NN \__xeCJK_boundary_group_math_brace: \scan_stop:
3462 \cs_new_protected:Npn \__xeCJK_boundary_group_math_peek:

```

```

3463 { \peek_after:Nw \__xeCJK_boundary_group_math_branches: }
3464 \cs_new_protected:Npn \__xeCJK_boundary_group_math_branches:
3465 {
3466   \token_if_math_toggle:NTF \l_peek_token
3467   { \__xeCJK_boundary_CJK_and_math: }
3468   {
3469     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3470     { \__xeCJK_boundary_reserve_space: }
3471     { \__xeCJK_boundary_group_end:n { CJK } }
3472   }
3473   \c_group_begin_token
3474 }
3475 % 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
3476 % 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。
3477 \cs_new:Npn \__xeCJK_boundary_identity:n #1 {#1}
3478 \cs_new_protected:Npn \__xeCJK_boundary_peek_math:w
3479 {
3480   \exp_after:wN \peek_after:Nw
3481   \exp_after:wN \__xeCJK_boundary_peek_math_branches:w
3482   \exp:w \exp_end_continue_f:w
3483 }
3484 \cs_new_protected:Npn \__xeCJK_boundary_peek_math_branches:w
3485 {
3486   \token_if_math_toggle:NTF \l_peek_token
3487   { \__xeCJK_boundary_CJK_and_math: }
3488   {
3489     \token_if_group_begin:NTF \l_peek_token
3490     { \__xeCJK_boundary_group_math:w }
3491     {
3492       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3493       { \__xeCJK_boundary_reserve_space: }
3494       { \__xeCJK_boundary_group_end:n { CJK } }
3495     }
3496   }
3497 }
3498 \cs_new_protected:Npn \__xeCJK_boundary_reserve_space:
3499 { \__xeCJK_boundary_group_end:n { CJK-space } }
3500 \cs_new_protected:Npn \__xeCJK_CJK_and_Boundary_relax:N #1
3501 {
3502   \__xeCJK_boundary_group_end:n { CJK }
3503   \token_if_eq_meaning:NNTF #1 \scan_stop:
3504   { #1 } { \cs_set_eq:NN #1 \scan_stop: #1 }
3505 }
3506 \cs_new_protected:Npn \__xeCJK_boundary_group_end:n #1
3507 {
3508   \group_align_safe_end:
3509   \xeCJK_class_group_end:
3510   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
3511   { \xeCJK_make_node:n { #1 } }
3512 }
\xeCJK_ignore_spaces:w 3513 \cs_new_protected:Npn \xeCJK_ignore_spaces:w
3514 {
3515   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
3516   {
3517     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3518     { \xeCJK_space_or_xecglue: } { \CJKecglue }
3519   }
3520   {
3521     \bool_if:NT \l__xeCJK_peek_ignore_spaces_bool
3522     {
3523       \dim_case:nn { \tex_lastkern:D }
3524       {
3525         { \__xeCJK_node:n { CJK } }
3526         { \xeCJK_remove_node: \xeCJK_make_node:n { CJK-space } }
3527         { \__xeCJK_node:n { default } }
3528         { \xeCJK_remove_node: \xeCJK_make_space_node: }
3529       }

```

```

3530         \group_align_safe_begin:
3531         \token_if_macro:NTF \l_peek_token
3532         { \__xeCJK_reserve_space_aux: }
3533         { \group_align_safe_end: }
3534     }
3535 }
3536 }
3537 \cs_new_protected:Npn \__xeCJK_reserve_space_aux:
3538 {
3539     \group_align_safe_end:
3540     \xeCJK_space_or_xecglue:
3541 }

3542 \xeCJK_inter_class_toks:nnn { CJK } { CJK }
3543 { \xeCJK_CJK_and_CJK:N }

\__xeCJK_CJK_and_CJK:N 3544 \cs_new_protected:Npn \xeCJK_CJK_and_CJK:N
3545 {
3546     \CJKglue
3547     \xeCJK_fallback_symbol:NN
3548     \CJKsymbol
3549 }

3550 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
3551 {
3552     \xeCJK_FullLeft_and_CJK:
3553     \xeCJK_fallback_symbol:NN
3554     \CJKsymbol
3555 }
3556 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
3557 {
3558     \xeCJK_FullRight_and_CJK:
3559     \xeCJK_fallback_symbol:NN
3560     \CJKsymbol
3561 }
3562 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
3563 {
3564     \clist_map_inline:nn { FullLeft , FullRight }
3565     {
3566         \xeCJK_inter_class_toks:nne {#1} {##1}
3567         { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
3568         \xeCJK_inter_class_toks:nne {##1} {#1}
3569         { \exp_not:c { xeCJK_##1_and_Default: } }
3570     }
3571 }

\l__xeCJK_halfright_nobreak_bool 3572 \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。

```

3573 \xeCJK_app_inter_class_toks:nnn { CJK } { HalfRight }
3574 {
3575     \bool_if:NT \l__xeCJK_halfright_nobreak_bool
3576     { \xeCJK_no_break: }
3577 }

```

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

```

3578 \xeCJK_inter_class_toks:nnn { FullRight } { HalfRight }
3579 {
3580     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3581     \xeCJK_class_group_end:
3582     \bool_if:NT \l__xeCJK_halfright_nobreak_bool
3583     { \xeCJK_no_break: }
3584     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3585 }

```



```

3586 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
3587 { \xeCJK_Boundary_and_FullLeft:N }
3588 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
3589 { \xeCJK_Boundary_and_FullRight:N }

3590 \xeCJK_inter_class_toks:nnn { FullLeft } { Boundary }
3591 { \xeCJK_FullLeft_and_Boundary: }
3592 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
3593 { \xeCJK_FullRight_and_Boundary: }

\xeCJK_FullLeft_and_Boundary: 3594 \cs_new_protected:Npn \xeCJK_FullLeft_and_Boundary:
3595 {
3596   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3597   {
3598     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3599     \xeCJK_class_group_end:
3600     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3601     \xeCJK_no_break:
3602     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3603     \__xeCJK_punct_boundary_guard:
3604   }
3605   {
3606     \xeCJK_class_group_end:
3607     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3608     \__xeCJK_nobreak_zero_glue:
3609   }
3610   \tex_ignorespaces:D
3611 }

\xeCJK_FullRight_and_Boundary: 3612 \cs_new_protected:Npn \xeCJK_FullRight_and_Boundary:
3613 {
3614   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3615   \xeCJK_class_group_end:
3616   \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3617   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3618   \__xeCJK_punct_boundary_guard:
3619   \tex_ignorespaces:D
3620 }

```

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）。

```

3621 \bool_new:N \g__xeCJK_par_guard_bool
3622 \dim_new:N \g__xeCJK_par_guard_dim
3623 \cs_new_protected:Npn \__xeCJK_punct_boundary_guard:
3624 {
3625   \mode_if_inner:TF
3626   { \tex_penalty:D \c_zero_int }
3627   {
3628     \bool_if:NT \g__xeCJK_punct_measure_fix_bool
3629     {
3630       \bool_gset_true:N \g__xeCJK_par_guard_bool
3631       \dim_gset:Nn \g__xeCJK_par_guard_dim { \tex_lastskip:D }
3632     }
3633   }
3634 }

```

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。

```

3635 \clist_map_inline:nn
3636 {
3637   Default , HalfLeft , HalfRight , NormalSpace ,
3638   CJK , FullLeft , FullRight , CM , HangulJamo ,

```

```

3639     HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
3640 }
3641 {
3642     \xeCJK_pre_inter_class_toks:nnn { Boundary } {#1}
3643     { \bool_gset_false:N \g__xeCJK_par_guard_bool }
3644 }

```

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。

```

3645 \AddToHook { para/begin }
3646 { \bool_gset_false:N \g__xeCJK_par_guard_bool }
3647 \AddToHook { para/end }
3648 {
3649     \bool_if:NT \g__xeCJK_par_guard_bool
3650     {
3651         \tex_kern:D \g__xeCJK_par_guard_dim
3652         \bool_gset_false:N \g__xeCJK_par_guard_bool
3653     }
3654 }

```

\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。

```

3655 \cs_new_protected:Npn \xeCJK_punct_node:N #1
3656 {
3657     \__xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
3658     \__xeCJK_make_node:N \l__xeCJK_tmp_dim
3659     \dim_set:Nn \l__xeCJK_tmp_dim { `#1 sp }
3660     \__xeCJK_make_node:N \l__xeCJK_tmp_dim
3661 }

```

__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”。

```

3662 \cs_new_protected:Npn \__xeCJK_punct_bound_unitization:NN #1#2
3663 {
3664     \dim_set:Nn #2
3665     {
3666         \dim_max:nn
3667         { \c_zero_dim }
3668         { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl #1 }
3669     }
3670     \dim_compare:nNnF {#2} < { 1pt }
3671     { \dim_set:Nn #2 { -1pt * \dim_ratio:nn {#2} { \c_max_dim } } }
3672 }

```

```

\xeCJK_punct_bound_kern:N 3673 \cs_new_protected:Npn \xeCJK_punct_bound_kern:N #1
__xeCJK_punct_bound_kern:NN 3674 {
3675     \exp_after:wN \__xeCJK_punct_bound_kern:NN
3676     \g__xeCJK_last_punct_tl #1
3677 }
3678 \cs_new_protected:Npn \__xeCJK_punct_bound_kern:NN #1#2
3679 {
3680     \xeCJK_get_punct_bounds:NN \l__xeCJK_aligni_tl #1
3681     \xeCJK_get_punct_kerning:NN #1 #2
3682     \__xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
3683     \skip_set:Nn \l__xeCJK_punct_kern_skip
3684     { \__xeCJK_use_dim_or_skip:nnn { bound_kern } #1 #2 }
3685     \dim_compare:nNnF \l__xeCJK_tmp_dim = \l__xeCJK_last_bound_dim
3686     { \__xeCJK_punct_bound_kern_ratio:NN #1 #2 }
3687     \bool_if:NTF \l__xeCJK_last_penalty_bool
3688     {
3689         \tex_penalty:D \l__xeCJK_last_penalty_int
3690         \skip_horizontal:N
3691     }
3692     { \__xeCJK_punct_bound_kern_aux:NNN #1 #2 }
3693     \l__xeCJK_punct_kern_skip
3694 }
3695 \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。

```

3696 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_ratio:NN #1#2
3697 {
3698   \dim_set:Nn \l__xeCJK_bound_dim
3699     { \_xeCJK_use_punct_dim:nNN { bound_width } #1 #2 }
3700   \dim_compare:nNnT \l__xeCJK_bound_dim > \c_zero_dim
3701     {
3702       \dim_compare:nNnF \l__xeCJK_last_bound_dim > \c_zero_dim
3703       {
3704         \dim_set:Nn \l__xeCJK_last_bound_dim
3705           {
3706             - \l__xeCJK_last_bound_dim *
3707               \dim_ratio:nn { \c_max_dim } { 1pt }
3708           }
3709       }
3710       \_xeCJK_punct_bound_kern_ratio_aux:N #2
3711     }
3712 }
3713 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_ratio_aux:N #1
3714 {
3715   \skip_set:Nn \l__xeCJK_punct_kern_skip
3716     {
3717       \l__xeCJK_punct_kern_skip *
3718       \dim_ratio:nn
3719         {
3720           \l__xeCJK_last_bound_dim
3721           + \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1
3722         }
3723       { \l__xeCJK_bound_dim }
3724     }
3725 }

\_xeCJK_nobreak_hskip:N 3726 \cs_new_protected:Npn \_xeCJK_nobreak_hskip:N
\_xeCJK_nobreak_hskip:n 3727 { \xeCJK_no_break: \skip_horizontal:N }
\_xeCJK_punct_bound_kern:N 3728 \cs_new_protected:Npn \_xeCJK_nobreak_hskip:n
\_xeCJK_punct_bound_breakable_kern:N 3729 { \xeCJK_no_break: \skip_horizontal:n }
3730 \cs_new_eq:NN \_xeCJK_punct_bound_kern:N \_xeCJK_nobreak_hskip:N
3731 \cs_new_protected:Npn \_xeCJK_punct_bound_breakable_kern:N
3732 {
3733   \tl_if_eq:NNTF \l__xeCJK_aligni_tl \c__xeCJK_right_tl
3734   {
3735     \tl_if_eq:NNTF \l__xeCJK_alignii_tl \c__xeCJK_left_tl
3736     { \skip_horizontal:N }
3737     { \_xeCJK_nobreak_hskip:N }
3738   }
3739   { \_xeCJK_nobreak_hskip:N }
3740 }
3741 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_aux:NNN #1#2
3742 {
3743   \str_if_eq:nnTF {#1} {#2}
3744   { \_xeCJK_nobreak_hskip:N }
3745   {
3746     \_xeCJK_punct_if_long:NNTF #1
3747     { \skip_horizontal:N }
3748     {
3749       \_xeCJK_punct_if_long:NNTF #2
3750       { \skip_horizontal:N }
3751       { \_xeCJK_punct_bound_kern:N }
3752     }
3753   }
3754 }

3755 \clist_map_inline:nn { CJK , FullLeft , FullRight }
3756 {
3757   \clist_map_inline:nn { FullLeft , FullRight }
3758   {
3759     \xeCJK_inter_class_toks:nne {#1} {##1}

```

```

3760      { \exp_not:c { xeCJK_#1_and_##1:N } }
3761    }
3762  }

```

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

```

3763 \cs_new_protected:Npn \_xeCJK_punct_bound_rule:NN #1#2
3764 {
3765   \tex_vrule:D
3766   width - \_xeCJK_use_punct_dim:nNN { bound } #1 #2 ~
3767   depth \c_zero_dim
3768   height \c_zero_dim \scan_stop:
3769 }

```

_xeCJK_punct_rule:NN 用于减少标点符号的左/右空白。

```

3770 \cs_new_protected:Npn \_xeCJK_punct_rule:NN #1#2
3771 {
3772   \tex_vrule:D
3773   width \_xeCJK_use_punct_dim:nNN { rule } #1 #2 ~
3774   depth \c_zero_dim
3775   height \c_zero_dim \scan_stop:
3776 }

```

_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。

```

3777 \cs_new_protected:Npn \_xeCJK_punct_glue:NN #1#2
3778 { \_xeCJK_punct_hskip:n { \_xeCJK_use_dim_or_skip:nNN { glue } #1 #2 } }
3779 \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。

```

3780 \cs_new_protected:Npn \xeCJK_punct_kern:NN #1#2
3781 {
3782   \str_if_eq:eeTF {#1} {#2}
3783   { \_xeCJK_punct_nobreak_kern:NN }
3784   {
3785     \bool_lazy_or:nnTF
3786     { \_xeCJK_punct_if_long_p:N #1 }
3787     { \_xeCJK_punct_if_long_p:N #2 }
3788     { \_xeCJK_punct_kern_break:NN #1 #2 }
3789     { \_xeCJK_punct_nobreak_kern:NN }
3790   }
3791   #1 #2
3792 }
3793 \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。

```

3794 \cs_new_protected:Npn \_xeCJK_punct_kern_break:NN #1#2
3795 {
3796   \exp_after:wN \_xeCJK_punct_if_right:NTF #1
3797   {
3798     \_xeCJK_punct_if_right:NTF #2
3799     {
3800       \bool_lazy_and:nnTF
3801       { \_xeCJK_punct_if_long_p:N #2 }
3802       { ! \_xeCJK_punct_if_nobreak_long_p:N #2 }
3803       { \_xeCJK_punct_breakable_kern:NN }

```

```

3804          { \__xeCJK_punct_nobreak_kern:NN }
3805      }
3806      { \__xeCJK_punct_breakable_kern:NN }
3807  }
3808  { \__xeCJK_punct_nobreak_kern:NN }
3809  }

```

```

\__xeCJK_punct_nobreak_kern:NN 3810 \cs_new_protected:Npn \__xeCJK_punct_nobreak_kern:NN #1#2
3811 { \__xeCJK_nobreak_hskip:n { \__xeCJK_use_dim_or_skip:nNN { kern } #1 #2 } }

```

```

\__xeCJK_punct_breakable_kern:NN 3812 \cs_new_protected:Npn \__xeCJK_punct_breakable_kern:NN #1#2
3813 {
3814   \exp_after:wN \__xeCJK_punct_if_right:NT #1
3815   { \__xeCJK_punct_rule:NN \c__xeCJK_right_tl #1 }
3816   \__xeCJK_punct_breakable_kern:n
3817   { \__xeCJK_use_dim_or_skip:nNN { bound_kern } #1 #2 }
3818   \__xeCJK_punct_if_right:NF #2
3819   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #2 }
3820 }
3821 \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。

```

3822 \tl_new:N \g__xeCJK_last_punct_tl

```

```

\xeCJK_FullLeft_and_CJK: 3823 \cs_new_protected:Npn \xeCJK_FullLeft_and_CJK:
3824 {
3825   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3826   {
3827     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3828     \xeCJK_no_break:
3829     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3830   }
3831   { }
3832   \__xeCJK_select_font:
3833 }

```

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

```

3834 \cs_new_protected:Npn \xeCJK_FullLeft_and_Default:
3835 {
3836   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3837   {
3838     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3839     \xeCJK_class_group_end: \xeCJK_no_break:
3840     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3841   }
3842   {
3843     \xeCJK_class_group_end:
3844     \__xeCJK_nobreak_zero_glue:
3845   }
3846 }
3847 \cs_new_protected:Npn \__xeCJK_nobreak_zero_glue:
3848 {
3849   \tex_penalty:D \c__xeCJK_nobreak_penalty_int
3850   \skip_horizontal:N \c_zero_skip
3851 }
3852 \cs_new_protected:Npn \__xeCJK_zero_glue:
3853 { \skip_horizontal:N \c_zero_skip }

```

```

\xeCJK_FullRight_and_CJK: 3854 \cs_new_protected:Npn \xeCJK_FullRight_and_CJK:
3855 {
3856   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3857   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3858   \__xeCJK_select_font:
3859   \CJKglue
3860 }

```

```

\XeCJK_FullRight_and_Default: 3861 \cs_new_protected:Npn \XeCJK_FullRight_and_Default:
3862 {
3863   \__XeCJK_punct_rule:NN \c__XeCJK_right_tl \g__XeCJK_last_punct_tl
3864   \XeCJK_class_group_end:
3865   \__XeCJK_punct_glue:NN \c__XeCJK_right_tl \g__XeCJK_last_punct_tl
3866 }

\XeCJK_Default_and_FullLeft:nN 3867 \cs_new_protected:Npn \XeCJK_Default_and_FullLeft:nN #1#2
3868 {
3869   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #2
3870   \__XeCJK_Default_and_FullLeft_glue:N #2
3871   \XeCJK_class_group_begin:
3872   \XeCJK_select_punct_font:
3873   \XeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
3874   \XeCJK_clear_Boundary_and_CJK_toks:
3875   \tl_gset:Nn \g__XeCJK_last_punct_tl {#2}
3876   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #2
3877   \XeCJK_fallback_punct_symbol:NN
3878   \CJKpunctsymbol #2
3879 }
3880 \cs_new_protected:Npn \__XeCJK_Default_and_FullLeft_glue:N #1
3881 { \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1 }

\XeCJK_CJK_and_FullLeft:N 3882 \cs_new_protected:Npn \XeCJK_CJK_and_FullLeft:N #1
3883 {
3884   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #1
3885   \__XeCJK_CJK_and_FullLeft_glue:N #1
3886   \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
3887   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #1
3888   \__XeCJK_select_punct_font:
3889   \XeCJK_fallback_punct_symbol:NN
3890   \CJKpunctsymbol #1
3891 }
3892 \cs_new_protected:Npn \__XeCJK_CJK_and_FullLeft_glue:N #1
3893 {
3894   \CJKglue
3895   \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1
3896 }

\XeCJK_Boundary_and_FullLeft:N 3897 \cs_new_protected:Npn \XeCJK_Boundary_and_FullLeft:N #1
3898 {
3899   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #1
3900   \__XeCJK_Boundary_and_FullLeft_glue:N #1
3901   \XeCJK_class_group_begin:
3902   \XeCJK_select_punct_font:
3903   \XeCJK_clear_Boundary_and_CJK_toks:
3904   \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
3905   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #1
3906   \XeCJK_fallback_punct_symbol:NN
3907   \CJKpunctsymbol #1
3908 }

```

__XeCJK_Boundary_and_FullLeft_glue:N 根据 \etex_lastnodetype:D 的值进行分别处理。

```

3909 \cs_new_protected:Npn \__XeCJK_Boundary_and_FullLeft_glue:N #1
3910 {
3911   \tl_set_eq:NN \l__XeCJK_alignii_tl \c__XeCJK_left_tl
3912   \group_begin: \exp_args:NNc \group_end: \cs_if_exist_use:NTF
3913   { \__XeCJK_bound_type_ \int_use:N \etex_lastnodetype:D _glue:Nn }
3914   {#1}
3915   { \use:n }
3916   { \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1 }
3917 }
3918 \tl_new:N \c__XeCJK_alignii_tl

```

__XeCJK_bound_type_-1_glue:Nn \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。

```

3919 \cs_new_protected:cpn { \__XeCJK_bound_type_-1 _glue:Nn } #1#2

```

```
3920 { \_xeCJK_zero_glue: }
```

`\_xeCJK_bound_type_1_glue:Nn` 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。

```
3921 \cs_new_protected:cpn { \_xeCJK_bound_type_ 1 _glue:Nn } #1
3922 {
3923   \int_do_while:nNnn \tex_lastnodetype:D = \c\_xeCJK_hlist_node
3924   { \_xeCJK_bound_hbox_auxi: }
3925   \_xeCJK_if_last_none:TF
3926   {
3927     \dim_case:nnF { \box_wd:N \l\_xeCJK_indent_box }
3928     {
3929       { \tex_parindent:D } { \_xeCJK_bound_hbox_auxii:nn }
3930       { \c_zero_dim }      { \use_i:nn }
3931     }
3932     { \use:nn }
3933   }
3934   { \use:nn }
3935   { \hbox_unpack_drop:N \l\_xeCJK_indent_box }
3936 }
3937 \cs_new_protected:Npn \_xeCJK_bound_hbox_auxi:
3938 {
3939   \box_set_to_last:N \l\_xeCJK_tmp_box
3940   \hbox_set:Nn \l\_xeCJK_indent_box
3941   {
3942     \box_use:N \l\_xeCJK_tmp_box
3943     \hbox_unpack:N \l\_xeCJK_indent_box
3944   }
3945 }
3946 \cs_new_protected:Npn \_xeCJK_bound_hbox_auxii:nn
3947 {
3948   \dim_compare:nNnTF
3949   { \box_ht:N \l\_xeCJK_tmp_box } = \c_zero_dim
3950   { \use_i:nn }
3951   { \use:nn }
3952 }
3953 \box_new:N \l\_xeCJK_indent_box
```

`\_xeCJK_bound_type_11_glue:Nn` 11 表示 glue node, 这里判断的目的是当全角左标点出现在 $\LaTeX$ 表格的非 p 列行首时, 能够对齐到单元格的边界。判断基于标准 $\LaTeX$ 表格的列格式 (`\@tabclassz`) 定义中, 在 l 列和 r 列前为了防止 `\tabcolsep` 被无意 `\unskip` 掉, 都加了 `\hskip1sp`, 而 c 列前则有 `\hfil`。`enumitem` 宏包修改了 `description` 环境中使用的 `\item(\enit@postlabel@i)`, 在这里起到影响作用的是 `\penalty\z@ \hskip\labelsep`。

```
3954 \cs_new_protected:cpn { \_xeCJK_bound_type_ 11 _glue:Nn } #1#2
3955 {
3956   \skip_if_finite:nTF { \tex_lastskip:D }
3957   { \_xeCJK_bound_glue_auxi:Nn #1 {#2} }
3958   { \_xeCJK_zero_glue: }
3959 }
3960 \cs_new_protected:Npn \_xeCJK_bound_glue_auxi:Nn #1#2
3961 {
3962   \_xeCJK_if_last_punct_glue:TF
3963   { \xeCJK_punct_bound_kern:N #1 }
3964   { \_xeCJK_bound_glue_auxii:n {#2} }
3965 }
3966 \cs_new_protected:Npn \_xeCJK_bound_glue_auxii:n #1
3967 {
3968   \skip_set_eq:NN \l\_xeCJK_last_skip \tex_lastskip:D
3969   \skip_if_eq:nnTF { \l\_xeCJK_last_skip } { 1sp }
3970   { \_xeCJK_zero_glue: }
3971   {
3972     \skip_if_eq:nnTF { \l\_xeCJK_last_skip } { \labelsep }
3973     {
3974       \tex_unskip:D
```

```

3975         \_xeCJK_if_last_penalty:TF
3976         {
3977             \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
3978             { \skip_horizontal:N \l__xeCJK_last_skip }
3979             { \skip_horizontal:N \l__xeCJK_last_skip #1 }
3980         }
3981         { \skip_horizontal:N \l__xeCJK_last_skip #1 }
3982     }
3983     {#1}
3984 }
3985 }

```

`\_xeCJK_bound_type_12_glue:Nn` 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 `\CJKglue`。

```

3986 \cs_new_protected:cpn { \_xeCJK_bound_type_12_glue:Nn } #1#2
3987 {
3988     \xeCJK_if_last_node:nF { CJK }
3989     { \xeCJK_if_last_node:nF { CJK-space } { \use_none:nn } }
3990     \xeCJK_remove_node: \CJKglue
3991     #2
3992 }

```

`\_xeCJK_bound_type_13_glue:n` 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 `\item` 后面时, 能对齐到边界。判断基于 `\item` 的内部定义 `@item` 对 `\everypar` 进行了修改, 在这里起到影响作用的是 `\box\@labels \penalty\z@`。以上判断都比较粗略, 暂时也没有想起更好的办法。

```

3993 \cs_new_protected:cpn { \_xeCJK_bound_type_13_glue:Nn } #1#2
3994 {
3995     \_xeCJK_if_last_punct_penalty:TF
3996     { \xeCJK_punct_bound_kern:N #1 }
3997     {
3998         \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
3999         {
4000             \tex_unpenalty:D
4001             \_xeCJK_if_last_hlist:TF
4002             { \tex_penalty:D \c_zero_int }
4003             { \tex_penalty:D \c_zero_int #2 }
4004         }
4005         {#2}
4006     }
4007 }

```

```

\xeCJK_Default_and_FullRight:nN 4008 \cs_new_protected:Npn \xeCJK_Default_and_FullRight:nN #1#2
4009 {
4010     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #2
4011     \_xeCJK_Default_and_FullRight_glue:N #2
4012     \xeCJK_class_group_begin:
4013     \xeCJK_select_punct_font:
4014     \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
4015     \xeCJK_clear_Boundary_and_CJK_toks:
4016     \tl_gset:Nn \g__xeCJK_last_punct_tl {#2}
4017     \xeCJK_FullRight_symbol:N #2
4018 }

```

```

\xeCJK_Boundary_and_FullRight:N 4019 \cs_new_protected:Npn \xeCJK_Boundary_and_FullRight:N #1
4020 {
4021     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4022     \xeCJK_if_last_punct:TF
4023     {
4024         \tl_set_eq:NN \l__xeCJK_alignii_tl \c__xeCJK_right_tl
4025         \xeCJK_punct_bound_kern:N
4026     }
4027     { \_xeCJK_Default_and_FullRight_glue:N }
4028     #1
4029     \xeCJK_class_group_begin:
4030     \xeCJK_select_punct_font:
4031     \xeCJK_clear_Boundary_and_CJK_toks:
4032     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}

```



```

4033 \xeCJK_FullRight_symbol:N #1
4034 }

```

```

\xeCJK_CJK_and_FullRight:N 4035 \cs_new_protected:Npn \xeCJK_CJK_and_FullRight:N #1
4036 {
4037   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4038   \__xeCJK_CJK_and_FullRight_glue:N #1
4039   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4040   \__xeCJK_select_punct_font:
4041   \xeCJK_FullRight_symbol:N #1
4042 }

```

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

```

4043 \cs_new_protected:Npn \xeCJK_if_last_punct:TF
4044 {
4045   \bool_set_false:N \l__xeCJK_last_penalty_bool
4046   \__xeCJK_if_last_glue:TF
4047   { \__xeCJK_if_last_punct_glue:TF }
4048   {
4049     \__xeCJK_if_last_penalty:TF
4050     { \__xeCJK_if_last_punct_penalty:TF }
4051     { \use_ii:nn }
4052   }
4053 }
4054 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue:TF
4055 {
4056   \prop_get:NpNTF \g__xeCJK_punct_skip_prop
4057   { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
4058   { \__xeCJK_if_last_punct_glue_auxi:TF }
4059   { \__xeCJK_if_last_punct_glue_auxii:TF }
4060 }
4061 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxi:TF
4062 {
4063   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
4064   \tex_unskip:D
4065   \int_compare:nNnTF \tex_lastpenalty:D = \c__xeCJK_nobreak_penalty_int
4066   { \__xeCJK_if_last_punct_auxi:TF { \use_i:nn } }
4067   {
4068     \xeCJK_if_last_node:TF
4069     { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
4070     { \use:n }
4071   }
4072   { \skip_horizontal:N \l__xeCJK_last_skip \use_ii:nn }
4073 }
4074 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxii:TF
4075 {
4076   \group_begin:
4077   \g__xeCJK_space_factor_int \tex_spacefactor:D
4078   \skip_if_eq:nnTF { \tex_lastskip:D } { \c_xeCJK_space_skip_tl }
4079   { \group_end: \__xeCJK_if_last_punct_glue_auxiii:TF }
4080   { \group_end: \use_ii:nn }
4081 }
4082 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxiii:TF
4083 {
4084   \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
4085   \tex_unskip:D
4086   \__xeCJK_if_last_glue:TF
4087   {
4088     \prop_get:NpNTF \g__xeCJK_punct_skip_prop
4089     { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
4090     { \__xeCJK_if_last_punct_glue_auxi:TF { \use_i:nn } }
4091     { \use:n }
4092   }
4093   { \use:n }
4094   { \skip_horizontal:N \l__xeCJK_tmp_skip \use_ii:nn }
4095 }

```

```

4096 \cs_new_protected:Npn \__xeCJK_if_last_punct_penalty:TF
4097 {
4098   \int_set_eq:NN \l__xeCJK_last_penalty_int \tex_lastpenalty:D
4099   \tex_unpenalty:D
4100   \bool_set_true:N \l__xeCJK_last_penalty_bool
4101   \__xeCJK_if_last_glue:TF
4102   { \__xeCJK_if_last_punct_glue:TF { \use_i:nn } }
4103   { \use:n }
4104   { \__xeCJK_last_punct_penalty_false:nn }
4105 }
4106 \cs_new_protected:Npn \__xeCJK_last_punct_penalty_false:nn #1#2
4107 {
4108   \bool_set_false:N \l__xeCJK_last_penalty_bool
4109   \tex_penalty:D \l__xeCJK_last_penalty_int
4110   #2
4111 }
4112 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxi:TF
4113 {
4114   \tex_unpenalty:D
4115   \bool_if:NF \l__xeCJK_last_penalty_bool
4116   {
4117     \bool_set_true:N \l__xeCJK_last_penalty_bool
4118     \int_set_eq:NN \l__xeCJK_last_penalty_int \c__xeCJK_nobreak_penalty_int
4119   }
4120   \xeCJK_if_last_node:TF
4121   { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
4122   { \use:n }
4123   { \xeCJK_no_break: \use_ii:nn }
4124 }
4125 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxii:TF
4126 {
4127   \dim_compare:nNnTF \l__xeCJK_last_kern_dim > \c_zero_dim
4128   { \__xeCJK_if_last_punct_auxiii:TF }
4129   { \__xeCJK_make_node:N \l__xeCJK_last_kern_dim \use_ii:nn }
4130 }
4131 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiii:TF
4132 {
4133   \int_case:nnTF { \tex_XeTeXcharclass:D \l__xeCJK_last_kern_dim }
4134   {
4135     { \xeCJK_class_num:n { FullRight } }
4136     { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_right_tl }
4137     { \xeCJK_class_num:n { FullLeft } }
4138     { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_left_tl }
4139   }
4140   { \__xeCJK_if_last_punct_auxiv:TF }
4141   { \use_ii:nn }
4142 }
4143 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiv:TF
4144 {
4145   \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
4146   \xeCJK_if_last_node:TF
4147   {
4148     \tl_gset:Ne \g__xeCJK_last_punct_tl
4149     { \tex_Uchar:D \l__xeCJK_tmp_dim }
4150     \dim_set_eq:NN \l__xeCJK_last_bound_dim \l__xeCJK_last_kern_dim
4151     \use_i:nn
4152   }
4153   { \__xeCJK_make_node:N \l__xeCJK_tmp_dim \use_ii:nn }
4154 }
4155 \tl_new:N \l__xeCJK_aligni_tl
4156 \tl_new:N \l__xeCJK_alignii_tl
4157 \int_new:N \l__xeCJK_last_penalty_int
4158 \dim_new:N \l__xeCJK_last_bound_dim
4159 \bool_new:N \l__xeCJK_last_penalty_bool
\xeCJK_if_last_node:TF 4160 \cs_new_protected:Npn \xeCJK_if_last_node:TF #1#2
4161 {
4162   \__xeCJK_if_last_kern:TF

```

```

4163     {
4164         \dim_set_eq:NN \l__xeCJK_last_kern_dim \tex_lastkern:D
4165         \tex_unkern:D
4166         \__xeCJK_if_last_kern:TF
4167         {
4168             \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_last_kern_dim }
4169             { \tex_unkern:D #1 }
4170             { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
4171         }
4172         { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
4173     }
4174     {#2}
4175 }
4176 \dim_new:N \l__xeCJK_last_kern_dim

```

```

\__xeCJK_CJK_and_FullRight_glue:N 4177 \cs_new_protected:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
\__xeCJK_Default_and_FullRight_glue:N 4178 {
4179     \__xeCJK_punct_if_long:NTF #1
4180     {
4181         \__xeCJK_punct_if_nobreak_long:NTF #1
4182         { \xeCJK_no_break: }
4183         { \xeCJK_allow_break: }
4184     }
4185     { \xeCJK_no_break: }
4186     \__xeCJK_punct_if_middle:NT #1
4187     {
4188         \CJKglue
4189         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
4190         \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
4191     }
4192 }
4193 \cs_new_protected:Npn \__xeCJK_Default_and_FullRight_glue:N #1
4194 {
4195     \__xeCJK_punct_if_long:NTF #1
4196     {
4197         \__xeCJK_punct_if_nobreak_long:NTF #1
4198         { \xeCJK_no_break: }
4199         { \xeCJK_allow_break: }
4200     }
4201     { \xeCJK_no_break: }
4202     \__xeCJK_punct_if_middle:NT #1
4203     {
4204         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
4205         \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
4206     }
4207 }

```

```

\xeCJK_FullLeft_and_FullLeft:N 4208 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullLeft:N #1
4209 {
4210     \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
4211     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4212     \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4213     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4214     \xeCJK_fallback_punct_symbol:NN
4215     \CJKpunctsymbol #1
4216 }

```

```

\xeCJK_FullLeft_and_FullRight:N 4217 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullRight:N #1
4218 {
4219     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4220     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4221     \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4222     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4223     \xeCJK_FullRight_symbol:N #1
4224 }

```

```

\xeCJK_FullRight_and_FullLeft:N 4225 \cs_new_protected:Npn \xeCJK_FullRight_and_FullLeft:N #1
4226 {

```

```

4227 \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
4228 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4229 \xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4230 \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4231 \xeCJK_fallback_punct_symbol:NN
4232 \CJKpunctsymbol #1
4233 }
\XeCJK_FullRight_and_FullRight:N 4234 \cs_new_protected:Npn \xeCJK_FullRight_and_FullRight:N #1
4235 {
4236 \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4237 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4238 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4239 \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4240 \xeCJK_FullRight_symbol:N #1
4241 }

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```

4242 \keys_define:nn { xeCJK / options }
4243 {
4244   CheckFullRight .choice: ,
4245   CheckFullRight / true .code:n =
4246   {
4247     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
4248     {
4249       \cs_set_eq:NN \__xeCJK_save_FullRight_check:
4250       \xeCJK_FullRight_and_Boundary:
4251       \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N
4252       \xeCJK_FullRight_symbol:N
4253       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
4254       \xeCJK_check_FullRight:
4255       \cs_set_eq:NN \xeCJK_FullRight_symbol:N
4256       \xeCJK_check_FullRight_symbol:Nw
4257     }
4258   } ,
4259   CheckFullRight / false .code:n =
4260   {
4261     \cs_if_eq:NNNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
4262     {
4263       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
4264       \__xeCJK_save_FullRight_check:
4265       \cs_set_eq:NN \xeCJK_FullRight_symbol:N
4266       \__xeCJK_save_FullRight_symbol:N
4267     }
4268   } ,
4269   CheckFullRight .default:n = { true }
4270 }

```

```

experiment/halfright-prebreakpenalty 4271 \keys_define:nn { xeCJK / boundary-register }
experiment/punct-measure-fix 4272 {
experiment/boundary-register 4273   command .code:n =
4274   { \__xeCJK_boundary_user_register_command:n {#1} } ,
4275   command .value_required:n = true ,
4276   strategy .code:n =
4277   { \__xeCJK_boundary_user_register_strategy:n {#1} } ,
4278   strategy .value_required:n = true ,
4279   mode .code:n =
4280   { \__xeCJK_boundary_user_register_mode:n {#1} } ,
4281   mode .value_required:n = true
4282 }
4283 \keys_define:nn { xeCJK / options / experiment }
4284 {
4285   halfright-prebreakpenalty .bool_set:N = \l__xeCJK_halfright_nobreak_bool ,
4286   halfright-prebreakpenalty .default:n = true ,

```

```

4287     halfright-prebreakpenalty .initial:n = false ,
4288     punct-measure-fix .bool_gset:N = \g__xeCJK_punct_measure_fix_bool ,
4289     punct-measure-fix .default:n = true ,
4290     punct-measure-fix .initial:n = false ,
4291     boundary-register .code:n =
4292     { \__xeCJK_boundary_user_register:n {#1} } ,
4293     boundary-register .value_required:n = true
4294 }

\__xeCJK_FullRight_symbol:N 4295 \cs_new_protected:Npn \xeCJK_FullRight_symbol:N
4296 {
4297     \xeCJK_fallback_punct_symbol:NN
4298     \CJKpunctsymbol
4299 }

\__xeCJK_check_FullRight: 4300 \cs_new_protected:Npn \xeCJK_check_FullRight:
4301 {
4302     \xeCJK_get_punct_bounds:No \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4303     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4304     \group_align_safe_begin:
4305     \token_case_meaning:NoTF \l_peek_token
4306     { \l__xeCJK_no_break_cs_case_tl }
4307     {
4308         \group_align_safe_end:
4309         \xeCJK_no_break:
4310         \group_insert_after:N \xeCJK_no_break:
4311     }
4312     { \group_align_safe_end: }
4313     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
4314     \xeCJK_class_group_end:
4315     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4316     \__xeCJK_punct_boundary_guard:
4317 }
4318 \prg_generate_conditional_variant:Nnn \token_case_meaning:Nn { No } { TF , F }

\__xeCJK_check_FullRight_symbol:Nw 4319 \cs_new_protected:Npn \xeCJK_check_FullRight_symbol:Nw #1
4320 { \peek_remove_spaces:n { \__xeCJK_save_FullRight_symbol:N #1 } }

\__xeCJK_cs_case_keys_define:nNnn 4321 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNnn #1#2#3#4#5
4322 {
4323     \tl_new:N #2
4324     \seq_new:N #3
4325     \keys_define:nn { xeCJK / options }
4326     {
4327         #1 .code:n =
4328         {
4329             \seq_set_split:Nnn #3 { } {##1}
4330             \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
4331         } ,
4332         #1+ .code:n =
4333         {
4334             \tl_map_inline:nn {##1}
4335             { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
4336             \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
4337         } ,
4338         #1- .code:n =
4339         {
4340             \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {####1} }
4341             \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
4342         }
4343     }
4344 }
4345 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:Nnn #1#2#3#4
4346 {
4347     \tl_clear:N #1
4348     \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
4349     #4
4350 }

```

NoBreakCS 设置不能在全角右标点之后断行的控制序列。

```
4351 \xeCJK_cs_case_keys_define:nNnn { NoBreakCS }
4352 \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。

```
4353 \NewDocumentCommand \xeCJKnobreak { }
4354 {
4355   \bool_set_true:N \l__xeCJK_tmp_bool
4356   \int_while_do:nNnn \tex_lastnodetype:D = \c__xeCJK_glue_node
4357   {
4358     \bool_if:NTF \l__xeCJK_tmp_bool
4359     {
4360       \bool_set_false:N \l__xeCJK_tmp_bool
4361       \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
4362     }
4363     { \skip_add:Nn \l__xeCJK_last_skip \tex_lastskip:D }
4364     \tex_unskip:D
4365   }
4366   \xeCJK_if_last_node:TF
4367   {
4368     \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
4369     \xeCJK_if_last_node:TF
4370     {
4371       \__xeCJK_if_last_glue:T
4372       {
4373         \exp_args:NNNo \tex_unskip:D \xeCJK_no_break:
4374         \skip_horizontal:n { \skip_use:N \tex_lastskip:D }
4375       }
4376       \__xeCJK_make_node:N \l__xeCJK_last_kern_dim
4377     }
4378     { }
4379     \__xeCJK_make_node:N \l__xeCJK_tmp_dim
4380   }
4381   { }
4382   \xeCJK_no_break:
4383   \bool_if:NF \l__xeCJK_tmp_bool
4384   { \skip_horizontal:N \l__xeCJK_last_skip }
4385 }
```

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```
4386 \keys_define:nn { xeCJK / options }
4387 {
4388   CheckSingle .choice: ,
4389   CheckSingle / true .code:n =
4390   {
4391     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4392     {
4393       \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
4394       \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4395     }
4396   } ,
4397   CheckSingle / false .code:n =
4398   {
4399     \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4400     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
4401   } ,
4402   CheckSingle .default:n = { true } ,
4403   CJKchecksingle .meta:n = { CheckSingle = true }
4404 }
```

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

```
4405 \keys_define:nn { xeCJK / options }
4406 {
4407   WidowPenalty .int_set:N = \l__xeCJK_widow_penalty_int ,
4408   WidowPenalty .default:n = { 10 000 }
4409 }
```

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

```
4410 \cs_new_protected:Npn \xeCJK_widow_penalty:
4411 { \tex_penalty:D \l__xeCJK_widow_penalty_int }
```

```
\xeCJK_check_single:Nw 4412 \cs_new_protected:Npn \xeCJK_check_single:Nw #1
\_xeCJK_check_single_end:N 4413 {
4414   \group_align_safe_begin:
4415   \peek_catcode:NTF \c_catcode_letter_token
4416   { \xeCJK_check_single:NNw #1 }
4417   {
4418     \token_if_other:NTF \l_peek_token
4419     { \xeCJK_check_single:NNw }
4420     { \_xeCJK_check_single_end:N }
4421     #1
4422   }
4423 }
4424 \cs_new_protected:Npn \_xeCJK_check_single_end:N
4425 {
4426   \group_align_safe_end:
4427   \_xeCJK_check_single_save:N
4428 }
```

`\xeCJK_check_single:NNw` 使用 `\group_align_safe_begin:` 和 `\group_align_safe_end:` 是为了防止在表格里报错。
`\_xeCJK_check_single_aux:nNNw`

```
4429 \cs_new_protected:Npn \xeCJK_check_single:NNw #1#2
4430 {
4431   \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
4432   {
4433     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
4434     {
4435       \bool_if:NTF \l__xeCJK_reserve_space_bool
4436       { \_xeCJK_check_single_end:N #1 #2 ~ }
4437       { \_xeCJK_check_single_space:NN #1#2 }
4438     }
4439     { \_xeCJK_check_single_end:N #1 #2 }
4440   }
4441   {
4442     \token_if_other:NTF \l_peek_token
4443     {
4444       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
4445       { \_xeCJK_check_single_space:NN }
4446       { \_xeCJK_check_single_end:N }
4447     }
4448     {
4449       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
4450       { \_xeCJK_check_single_aux:nNNw { ~ } }
4451       { \_xeCJK_check_single_aux:nNNw { } }
4452     }
4453     #1 #2
4454   }
4455 }
4456 \cs_new_protected:Npn \_xeCJK_check_single_aux:nNNw #1#2#3
4457 {
4458   \token_if_cs:NTF \l_peek_token
4459   { \xeCJK_check_single_cs:NNn }
4460   { \xeCJK_check_single_end:NNnw }
4461   #2 #3 {#1}
4462 }
```

```

\XeCJK_check_single_end:NNnw 4463 \cs_new_protected:Npn \__XeCJK_check_single_end_aux:NNn #1#2#3
\__XeCJK_check_single_end_aux:NNn 4464 { \__XeCJK_check_single_end:N #1 #2 #3 }
\__XeCJK_check_single_end_equation:NNnw 4465 \cs_new_eq:NN \XeCJK_check_single_end:NNnw \__XeCJK_check_single_end_aux:NNn
4466 \cs_new_protected:Npn \__XeCJK_check_single_end_equation:NNnw
4467 {
4468   \token_if_math_toggle:NTF \l_peek_token
4469   { \XeCJK_check_single_equation:NNnNw }
4470   { \__XeCJK_check_single_end_aux:NNn }
4471 }

PlainEquation 4472 \keys_define:nn { XeCJK / options }
4473 {
4474   PlainEquation .choice: ,
4475   PlainEquation / true .code:n =
4476   {
4477     \cs_set_eq:NN \XeCJK_check_single_end:NNnw
4478     \__XeCJK_check_single_end_equation:NNnw
4479   } ,
4480   PlainEquation / false .code:n =
4481   {
4482     \cs_set_eq:NN \XeCJK_check_single_end:NNnw
4483     \__XeCJK_check_single_end_aux:NNn
4484   } ,
4485   PlainEquation .default:n = { true } ,
4486 }

\__XeCJK_check_single_space:NN 4487 \cs_new_protected:Npn \__XeCJK_check_single_space:NN #1#2
4488 {
4489   \XeCJK_if_CJK_class:NTF #2
4490   {
4491     \XeCJK_if_CJK_class:NTF \l_peek_token
4492     { \__XeCJK_check_single_end:N #1 #2 }
4493     { \__XeCJK_check_single_end:N #1 #2 ~ }
4494   }
4495   { \__XeCJK_check_single_end:N #1 #2 ~ }
4496 }

\XeCJK_check_single_equation:NNnNw 4497 \cs_new_protected:Npn \XeCJK_check_single_equation:NNnNw #1#2#3#4
4498 {
4499   \peek_catcode:NTF \c_math_toggle_token
4500   {
4501     \XeCJK_widow_penalty: \__XeCJK_check_single_end:N #1
4502     \XeCJK_make_node:n { CJK-widow } #2 #4
4503   }
4504   { \__XeCJK_check_single_end:N #1 #2#3#4 }
4505 }

```

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

```

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

```

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

```

4506 \cs_new_protected:Npn \XeCJK_check_single_cs:NNn #1#2#3
4507 {
4508   \token_case_meaning:Nof \l_peek_token
4509   { \l__XeCJK_check_single_cs_case_tl }
4510   { \use_iii:nnn }
4511   { \XeCJK_check_single_env:nnNn }
4512   {
4513     \XeCJK_widow_penalty:
4514     \__XeCJK_check_single_end:N #1

```



```

4515         \xeCJK_make_node:n { CJK-widow } #2#3
4516     }
4517     { \_xeCJK_check_single_end:N #1 #2#3 }
4518 }
4519 \tl_new:N \l__xeCJK_check_single_cs_case_tl

\xeCJK_check_single_env:nnNn 4520 \cs_new_protected:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
4521 {
4522     \str_case_e:noTF {#4}
4523     { \l__xeCJK_inline_env_case_tl }
4524     {#2}
4525     {#1}
4526     #3 {#4}
4527 }
4528 \prg_generate_conditional_variant:Nnn \str_case_e:nn { no } { TF }

NewLineCS 4529 \xeCJK_cs_case_keys_define:nnNnn { NewLineCS }
4530 \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
4531 { \use_ii:nnn }
4532 {
4533     \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
4534     \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
4535 }

EnvCS 4536 \xeCJK_cs_case_keys_define:nnNnn { EnvCS }
4537 \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
4538 { \use:n }
4539 {
4540     \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
4541     \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
4542 }

InlineEnv 4543 \keys_define:nn { xeCJK / options }
4544 {
4545     InlineEnv .code:n =
4546     {
4547         \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
4548         \__xeCJK_update_inline_env_case_tl:
4549     } ,
4550     InlineEnv+ .code:n =
4551     {
4552         \clist_map_inline:nn {#1}
4553         {
4554             \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
4555             { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
4556         }
4557         \__xeCJK_update_inline_env_case_tl:
4558     } ,
4559     InlineEnv- .code:n =
4560     {
4561         \clist_map_inline:nn {#1}
4562         { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
4563         \__xeCJK_update_inline_env_case_tl:
4564     }
4565 }
4566 \seq_new:N \l__xeCJK_inline_env_seq

\_xeCJK_update_inline_env_case_tl: 4567 \cs_new_protected:Npn \_xeCJK_update_inline_env_case_tl:
4568 {
4569     \tl_clear:N \l__xeCJK_inline_env_case_tl
4570     \seq_map_inline:Nn \l__xeCJK_inline_env_seq
4571     { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { } } }
4572 }
4573 \tl_new:N \l__xeCJK_inline_env_case_tl

```

5.9 增加 CJK 子分区

```

\g__xeCJK_CJK_sub_class_seq 4574 \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。

4575 \NewDocumentCommand \XeCJKDeclareSubCJKBlock
4576 { s > { \TrimSpaces } m m }
4577 {
4578   \XeCJK_declare_sub_char_class:nen { CJK } {#2} {#3}
4579   \IfBooleanT {#1} { \XeCJKResetPunctClass }
4580 }
4581 \@onlypreamble \XeCJKDeclareSubCJKBlock

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

4582 \bool_new:N \l__xeCJK_sub_cancel_bool
4583 \NewDocumentCommand \XeCJKCancelSubCJKBlock { s m }
4584 {
4585   \bool_if:NF \l__xeCJK_sub_cancel_bool
4586   {
4587     \bool_set_true:N \l__xeCJK_sub_cancel_bool
4588     \__xeCJK_sub_restore_or_cancel:e {#2}
4589     \IfBooleanT {#1} { \XeCJKResetPunctClass }
4590   }
4591 }
4592 \NewDocumentCommand \XeCJKRestoreSubCJKBlock { s m }
4593 {
4594   \bool_if:NT \l__xeCJK_sub_cancel_bool
4595   {
4596     \bool_set_false:N \l__xeCJK_sub_cancel_bool
4597     \__xeCJK_sub_restore_or_cancel:e {#2}
4598     \IfBooleanT {#1} { \XeCJKResetPunctClass }
4599   }
4600 }

\__xeCJK_sub_restore_or_cancel:n 4601 \cs_new_protected:Npn \__xeCJK_sub_restore_or_cancel:n #1
4602 {
4603   \clist_map_inline:nn {#1}
4604   {
4605     \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
4606     {
4607       \XeCJK_declare_char_class:nn
4608       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
4609       { \use:c { g__xeCJK_CJK/##1_range_clist } }
4610     }
4611     { \__xeCJK_error:ne { SubBlock-undefined } {##1} }
4612   }
4613 }
4614 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { e }
4615 \__xeCJK_msg_new:nn { SubBlock-undefined }
4616 {
4617   The~CJK~sub~block~`#1'~is~undefined.\\
4618   Try~to~use~\token_to_str:N \XeCJKDeclareSubCJKBlock \
4619   to~declare~it.
4620 }

\XeCJK_declare_sub_char_class:nnn 4621 \cs_new_protected:Npn \XeCJK_declare_sub_char_class:nnn #1#2#3
4622 {
4623   \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
4624   {
4625     \XeCJK_new_class:n { #1/#2 }
4626     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
4627     \XeCJK_new_sub_key:n {#2}
4628   }
4629   \XeCJK_declare_char_class:nn { #1/#2 } {#3}
4630 }
4631 \cs_generate_variant:Nn \XeCJK_declare_sub_char_class:nnn { ne }

```

```

4632 \cs_new_protected:Npn \__xeCJK_set_sub_class_toks:nn #1#2
4633 {
4634   \seq_map_inline:Nn \g__xeCJK_base_class_seq
4635   {
4636     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
4637     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
4638     \str_if_eq:nnTF {##1} { CJK }
4639     {
4640       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
4641       { \__xeCJK_switch_font:nn {#1} {#2} }
4642     }
4643     {
4644       \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
4645       { \xeCJK_fallback_symbol:NN }
4646       {
4647         \__xeCJK_switch_font:nn {#1} {#2}
4648         \xeCJK_fallback_symbol:NN
4649       }
4650     }
4651   }
4652   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
4653   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
4654   {
4655     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
4656     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
4657     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
4658     { \__xeCJK_switch_font:nn {#2} {##1} }
4659     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
4660     { \__xeCJK_switch_font:nn {##1} {#2} }
4661   }
4662   \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
4663   \__xeCJK_save_CJK_class:n { #1/#2 }
4664   \clist_map_inline:nn
4665   {
4666     CJK , FullLeft , FullRight , HangulJamo ,
4667     HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
4668   }
4669   {
4670     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
4671     { \__xeCJK_switch_font:nn {#2} {#1} }
4672   }
4673 }

```

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。

```

4674 \cs_if_exist:NF \tex_XeTeXglyphbounds:D
4675 {
4676   \__xeCJK_msg_new:nn { XeTeX-too-old }
4677   {
4678     \token_to_str:N \tex_XeTeXglyphbounds:D \ is~not~defined.\\
4679     CJK~punctuation~kerning~will~not~be~available.\\
4680     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
4681   }
4682   \__xeCJK_error:n { XeTeX-too-old }
4683   \AtEndOfPackage
4684   {
4685     \keys_define:nn { xeCJK / options }
4686     {
4687       PunctStyle .code:n =
4688       { \__xeCJK_error:ne { punct-style-unknown } {#1} }
4689     }
4690     \seq_gclear:N \g__xeCJK_punct_style_seq
4691     \__xeCJK_set_punct_style:n { plain }

```

```

4692     }
4693 }

```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```

4694 \NewDocumentCommand \xeCJKsetwidth { s m m }
4695 {
4696   \IfBooleanTF {#1}
4697   {
4698     \tl_map_inline:en {#2}
4699     {
4700       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
4701       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
4702       \tl_gset:cn { g__xeCJK_punct_bound_width/##1/tl } {#3}
4703     }
4704   }
4705   {
4706     \tl_map_inline:en {#2}
4707     {
4708       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
4709       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
4710       \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#3}
4711     }
4712   }
4713 }
4714 \@onlypreamble \xeCJKsetwidth
4715 \cs_generate_variant:Nn \tl_map_inline:nn { e }

```

\xeCJKsetkern 手动设置相邻标点的距离。

```

4716 \NewDocumentCommand \xeCJKsetkern { m m m }
4717 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
4718 \@onlypreamble \xeCJKsetkern

\c__xeCJK_left_tl 4719 \tl_const:Nn \c__xeCJK_left_tl { left }
\c__xeCJK_right_tl 4720 \tl_const:Nn \c__xeCJK_right_tl { right }

```

AllowBreakBetweenPuncts 相关选项声明。

```

KaiMingPunct
LongPunct
NoBreakLongPunct
MiddlePunct
PoZheHaoLigature
LatinPunct
CJLineBreak
PunctWidth
PunctBoundWidth
RubberPunctSkip

4721 \keys_define:nn { xeCJK / options }
4722 {
4723   AllowBreakBetweenPuncts .choice: ,
4724   AllowBreakBetweenPuncts / true .code:n =
4725   {
4726     \bool_set_true:N \l__xeCJK_punct_breakable_bool
4727     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
4728     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
4729     \__xeCJK_punct_bound_breakable_kern:N
4730   } ,
4731   AllowBreakBetweenPuncts / false .code:n =
4732   {
4733     \bool_set_false:N \l__xeCJK_punct_breakable_bool
4734     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_kern:NN
4735     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
4736     \__xeCJK_nobreak_hskip:N
4737   } ,
4738   AllowBreakBetweenPuncts .default:n = { true } ,
4739   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
4740   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
4741   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
4742   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
4743   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
4744   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
4745   NoBreakLongPunct .code:n = { \__xeCJK_set_special_punct:nn { nobreak_long } {#1} } ,
4746   NoBreakLongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { nobreak_long } {#1} } ,
4747   NoBreakLongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { nobreak_long } {#1} } ,
4748   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
4749   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,

```

```

4750 MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
4751 PoZheHaoLigature .choice: ,
4752 PoZheHaoLigature / true .code:n =
4753 {
4754     \bool_set_true:N \l__xeCJK_pozhehao_ligature_bool
4755     \xeCJK_declare_char_class:nn { PoZheHao } { "2014 , "2015 }
4756 } ,
4757 PoZheHaoLigature / false .code:n =
4758 {
4759     \bool_set_false:N \l__xeCJK_pozhehao_ligature_bool
4760     \xeCJK_declare_char_class:nn { FullRight } { "2014 }
4761     \xeCJK_declare_char_class:nn { Default } { "2015 }
4762 } ,
4763 PoZheHaoLigature .default:n = { true } ,
4764 LatinPunct .choice: ,
4765 LatinPunct / true .code:n =
4766 {
4767     \bool_set_true:N \l__xeCJK_latin_punct_bool
4768     \xeCJK_declare_char_class:nn { HalfLeft }
4769     { "2018 , "201C }
4770     \xeCJK_declare_char_class:nn { HalfRight }
4771     { "00B7 , "2019 , "201D , "2025 , "2026 , "2027 }
4772 } ,
4773 LatinPunct / false .code:n =
4774 {
4775     \bool_set_false:N \l__xeCJK_latin_punct_bool
4776     \xeCJK_declare_char_class:nn { FullLeft }
4777     { "2018 , "201C }
4778     \xeCJK_declare_char_class:nn { FullRight }
4779     { "00B7 , "2019 , "201D , "2025 , "2026 , "2027 }
4780 } ,
4781 LatinPunct .default:n = { true } ,
4782 CJLineBreak .choices:nn = { normal , strict }
4783 {
4784     \str_if_eq:nnTF {#1} { strict }
4785     {
4786         \bool_set_true:N \l__xeCJK_CJ_strict_bool
4787         \xeCJK_declare_char_class:nN { CJStarter } \c__xeCJK_CJ_chars_clist
4788     }
4789     {
4790         \bool_set_false:N \l__xeCJK_CJ_strict_bool
4791         \xeCJK_declare_char_class:nN { CJK } \c__xeCJK_CJ_chars_clist
4792     }
4793 } ,
4794 CJLineBreak .initial:n = { normal } ,
4795 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
4796 PunctBoundWidth.tl_gset:N = \g__xeCJK_punct_bound_width_tl ,
4797 PunctWidth .value_required:n = true ,
4798 PunctBoundWidth .value_required:n = true ,
4799 RubberPunctSkip .choice: ,
4800 RubberPunctSkip .default:n = { true } ,
4801 RubberPunctSkip / true .code:n =
4802 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN } ,
4803 RubberPunctSkip / plus .code:n =
4804 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_plus:nNN } ,
4805 RubberPunctSkip / minus .code:n =
4806 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_minus:nNN } ,
4807 RubberPunctSkip / false .code:n =
4808 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_dim:nNN }
4809 }
4810 \bool_new:N \l__xeCJK_punct_breakable_bool

```

相关选项定义的辅助函数。

```

4811 \clist_new:N \g__xeCJK_special_punct_clist
4812 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , nobreak_long , middle }
4813 \cs_new:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
4814 \cs_new:Npn \__xeCJK_special_punct_tl:n #1#2 { g__xeCJK_special_punct_#1_#2_tl }

```

```

4815 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
4816 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
4817 \cs_new_protected:Npn \__xeCJK_set_special_punct:nn #1#2
4818 {
4819   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
4820   { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
4821   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
4822   \tl_map_inline:en {#2}
4823   {
4824     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4825     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4826   }
4827 }
4828 \cs_new_protected:Npn \__xeCJK_add_special_punct:nn #1#2
4829 {
4830   \tl_map_inline:en {#2}
4831   {
4832     \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
4833     {
4834       \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4835       \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4836     }
4837   }
4838 }
4839 \cs_new_protected:Npn \__xeCJK_sub_special_punct:nn #1#2
4840 {
4841   \tl_map_inline:en {#2}
4842   {
4843     \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4844     \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4845   }
4846 }

```

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。

```

4847 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
4848 {
4849   \if_int_compare:w \xeCJK_token_value_class:N #1 =
4850     \xeCJK_class_num:n { FullRight }
4851     \prg_return_true:
4852   \else:
4853     \if_int_compare:w \xeCJK_token_value_class:N #1 =
4854       \xeCJK_class_num:n { PoZheHao }
4855       \prg_return_true:
4856     \else:
4857       \prg_return_false:
4858     \fi:
4859   \fi:
4860 }
4861 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
4862 {
4863   \exp_args:Nc
4864   \prg_new_conditional:Npnn { \__xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
4865   {
4866     \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
4867     \prg_return_true: \else: \prg_return_false: \fi:
4868   }
4869 }

```

一些用于记录的辅助函数。

```

4870 \cs_new:Npn \__xeCJK_punct_csname:n #1
4871 { c__xeCJK_\l_xeCJK_current_punct_font_tl/\l_xeCJK_punct_style_tl/#1/tl }
4872 \cs_new:Npn \__xeCJK_use_punct_dim:nN #1#2
4873 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2 } } }
4874 \cs_new:Npn \__xeCJK_use_punct_dim:nNN #1#2#3
4875 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2/#3 } } }

```

```

4876 \cs_new:Npn \__xeCJK_use_punct_skip:nNN #1#2#3
4877 { \use:c { \__xeCJK_punct_csname:n { skip/#1/#2/#3 } } }
4878 \cs_new:Npn \__xeCJK_use_punct_skip_plus:nNN #1#2#3
4879 { \use:c { \__xeCJK_punct_csname:n { skip/plus/#1/#2/#3 } } }
4880 \cs_new:Npn \__xeCJK_use_punct_skip_minus:nNN #1#2#3
4881 { \use:c { \__xeCJK_punct_csname:n { skip/minus/#1/#2/#3 } } }
4882 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNn #1#2
4883 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2 } }
4884 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNNn #1#2#3
4885 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2/#3 } }
4886 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNn #1#2#3#4
4887 {
4888   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { #1/#2/#3 } {#4}
4889   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { plus/#1/#2/#3 } {#4}
4890   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { minus/#1/#2/#3 } {#4}
4891 }
4892 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNnnn #1#2#3#4#5#6
4893 {
4894   \exp_last_unbraced:Ne
4895   \__xeCJK_save_punct_skip_aux:nnnnn
4896   {
4897     {#1}
4898     { #1/#2/#3 }
4899     { \dim_eval:n {#4} }
4900     { \dim_max:nn { \c_zero_dim } {#5} }
4901     { \dim_max:nn { \c_zero_dim } {#6} }
4902   }
4903 }
4904 \cs_new_protected:Npn \__xeCJK_save_punct_skip_aux:nnnnn #1#2#3#4#5
4905 {
4906   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4907   {#2} { #3 ~ plus ~ #4 ~ minus ~ #5 ~ }
4908   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4909   { plus/#2 } { #3 ~ plus ~ #4 ~ }
4910   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4911   { minus/#2 } { #3 ~ minus ~ #5 ~ }
4912 }
4913 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:nnnn #1#2#3#4
4914 {
4915   \__xeCJK_save_punct_width_aux:cen
4916   { \__xeCJK_punct_csname:n { #1/#3 } }
4917   { \use:c { #1_eval:n } {#4} }
4918   {#2}
4919 }
4920 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:Nnn #1#2#3
4921 {
4922   \tl_const:Nn #1 {#2}
4923   \str_if_eq:nnT {#3} { glue }
4924   { \prop_gput:Nnn \g__xeCJK_punct_skip_prop {#2} { } }
4925 }
4926 \prop_new:N \g__xeCJK_punct_skip_prop
4927 \prop_gput:Non \g__xeCJK_punct_skip_prop { \skip_use:N \c_zero_skip } { }
4928 \cs_generate_variant:Nn \__xeCJK_save_punct_width_aux:Nnn { ce }
4929 \cs_new_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN

```

定义标点处理模板。

```

4930 \DeclareObjectType { xeCJK / punctuation } { 0 }
4931 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { 0 }
4932 {
4933   enabled-global-setting      : boolean = true ,
4934   fixed-punct-width          : length  = \c_max_dim ,
4935   fixed-punct-ratio          : real    = \c_one_fp ,
4936   mixed-punct-width          : length  = \KeyValue { fixed-punct-width } ,
4937   mixed-punct-ratio          : real    = \KeyValue { fixed-punct-ratio } ,
4938   middle-punct-width         : length  = \KeyValue { fixed-punct-width } ,
4939   middle-punct-ratio         : real    = \KeyValue { fixed-punct-ratio } ,

```

```

4940     fixed-margin-width           : length = \c_max_dim ,
4941     fixed-margin-ratio           : real   = \c_one_fp ,
4942     mixed-margin-width           : length = \KeyValue { fixed-margin-width } ,
4943     mixed-margin-ratio           : real   = \KeyValue { fixed-margin-ratio } ,
4944     middle-margin-width          : length = \KeyValue { fixed-margin-width } ,
4945     middle-margin-ratio          : real   = \KeyValue { fixed-margin-ratio } ,
4946     bound-punct-width            : length = \c_max_dim ,
4947     bound-punct-ratio            : real   = \c_nan_fp ,
4948     bound-margin-width           : length = \c_max_dim ,
4949     bound-margin-ratio           : real   = \c_zero_fp ,
4950     enabled-hanging              : boolean = false ,
4951     add-min-bound-to-margin      : boolean = false ,
4952     optimize-margin              : boolean = false ,
4953     margin-minimum               : length = \c_zero_dim ,
4954     enabled-kerning              : boolean = true ,
4955     enabled-left-right-kerning   : boolean = true ,
4956     min-bound-to-kerning         : boolean = false ,
4957     kerning-total-width          : length = \c_max_dim ,
4958     kerning-total-ratio          : real   = 0.75 ,
4959     optimize-kerning             : boolean = false ,
4960     same-align-margin            : length = \c_max_dim ,
4961     same-align-ratio             : real   = \c_nan_fp ,
4962     different-align-margin       : length = \c_max_dim ,
4963     different-align-ratio        : real   = \c_nan_fp ,
4964     kerning-margin-width         : length = \c_max_dim ,
4965     kerning-margin-ratio         : real   = \c_one_fp ,
4966     kerning-margin-minimum       : length = \c_zero_dim
4967 }

4968 \DeclareTemplateCode { xeCJK / punctuation } { basic } { 0 }
4969 {
4970     enabled-global-setting        = \l__xeCJK_enabled_global_setting_bool ,
4971     fixed-punct-width            = \l__xeCJK_fixed_punct_width_dim ,
4972     fixed-punct-ratio            = \l__xeCJK_fixed_punct_ratio_fp ,
4973     mixed-punct-width            = \l__xeCJK_mixed_punct_width_dim ,
4974     mixed-punct-ratio            = \l__xeCJK_mixed_punct_ratio_fp ,
4975     middle-punct-width           = \l__xeCJK_middle_punct_width_dim ,
4976     middle-punct-ratio           = \l__xeCJK_middle_punct_ratio_fp ,
4977     fixed-margin-width           = \l__xeCJK_fixed_margin_width_dim ,
4978     fixed-margin-ratio           = \l__xeCJK_fixed_margin_ratio_fp ,
4979     mixed-margin-width           = \l__xeCJK_mixed_margin_width_dim ,
4980     mixed-margin-ratio           = \l__xeCJK_mixed_margin_ratio_fp ,
4981     middle-margin-width          = \l__xeCJK_middle_margin_width_dim ,
4982     middle-margin-ratio          = \l__xeCJK_middle_margin_ratio_fp ,
4983     bound-punct-width            = \l__xeCJK_bound_punct_width_dim ,
4984     bound-punct-ratio            = \l__xeCJK_bound_punct_ratio_fp ,
4985     bound-margin-width           = \l__xeCJK_bound_margin_width_dim ,
4986     bound-margin-ratio           = \l__xeCJK_bound_margin_ratio_fp ,
4987     enabled-hanging              = \l__xeCJK_enabled_hanging_bool ,
4988     add-min-bound-to-margin      = \l__xeCJK_add_min_bound_to_margin_bool ,
4989     optimize-margin              = \l__xeCJK_optimize_margin_bool ,
4990     margin-minimum               = \l__xeCJK_margin_minimum_dim ,
4991     enabled-kerning              = \l__xeCJK_enabled_kerning_bool ,
4992     enabled-left-right-kerning   = \l__xeCJK_enabled_left_right_kerning_bool ,
4993     min-bound-to-kerning         = \l__xeCJK_min_bound_to_kerning_bool ,
4994     kerning-total-width          = \l__xeCJK_kerning_total_width_dim ,
4995     kerning-total-ratio          = \l__xeCJK_kerning_total_ratio_fp ,
4996     optimize-kerning             = \l__xeCJK_optimize_kerning_bool ,
4997     same-align-margin            = \l__xeCJK_same_align_margin_dim ,
4998     same-align-ratio             = \l__xeCJK_same_align_ratio_fp ,
4999     different-align-margin       = \l__xeCJK_different_align_margin_dim ,
5000     different-align-ratio        = \l__xeCJK_different_align_ratio_fp ,
5001     kerning-margin-width         = \l__xeCJK_kerning_margin_width_dim ,
5002     kerning-margin-ratio         = \l__xeCJK_kerning_margin_ratio_fp ,
5003     kerning-margin-minimum       = \l__xeCJK_kerning_margin_minimum_dim
5004 }
5005 { \AssignTemplateKeys }

```


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

```

5006 \cs_new_protected:Npn \xeCJK_get_punct_bounds:NN #1#2
5007 {
5008   \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/glue/#1/#2 } }
5009   { \__xeCJK_get_punct_bounds_aux:NN #1 #2 }
5010 }
5011 \cs_new_protected:Npn \__xeCJK_get_punct_bounds_aux:NN
5012 {
5013   \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
5014   { \__xeCJK_save_punct_margin_plain:NN }
5015   { \__xeCJK_save_punct_margin:NN }
5016 }
5017 \cs_new_protected:Npn \xeCJK_get_punct_bounds:No
5018 { \exp_last_unbraced:NNo \xeCJK_get_punct_bounds:NN }
5019 \cs_new_protected:Npn \__xeCJK_save_punct_margin_plain:NN #1#2
5020 {
5021   \__xeCJK_save_punct_dim:nNNn { glue } #1 #2 { \c_zero_dim }
5022   \__xeCJK_save_punct_dim:nNNn { offset } #1 #2 { \c_zero_dim }
5023   \__xeCJK_save_punct_dim:nNNn { margin } #1 #2 { \c_zero_dim }
5024   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_left_tl {#2} { \c_zero_dim }
5025   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_right_tl {#2} { \c_zero_dim }
5026   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
5027   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
5028   \__xeCJK_save_punct_skip:nNNn { glue } #1 #2 { \c_zero_skip }
5029 }
5030 \cs_new_protected:Npn \__xeCJK_save_punct_margin:NN #1#2
5031 {
5032   \group_begin:
5033   \xeCJK_select_punct_font:
5034   \xeCJK_fallback_punct_symbol:NN
5035   \xeCJK_calc_punct_dimen:N #2
5036   \group_end:
5037   \dim_set:Nn \l__xeCJK_bound_dim
5038   { \__xeCJK_use_punct_dim:nNN { bound } #1 #2 }
5039   \tl_if_eq:NNTF #1 \c__xeCJK_right_tl
5040   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_left_tl }
5041   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_right_tl }
5042   \dim_set:Nn \l__xeCJK_reverse_bound_dim
5043   { \__xeCJK_use_punct_dim:nNN { bound } \l__xeCJK_reverse_tl #2 }
5044   \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
5045   \xeCJK_punct_margin_process:NN #1 #2
5046   \xeCJK_punct_offset_process:NN #1 #2
5047   \__xeCJK_punct_if_long:NT #2
5048   { \__xeCJK_long_punct_kerning:N #2 }
5049 }
5050 \tl_new:N \l__xeCJK_reverse_tl
5051 \dim_new:N \l__xeCJK_bound_dim
5052 \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);
- dimen + width - 2F, 用于字面溢出字框的字体(如方正兰亭黑);
- 2width - 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。

```

5053 \cs_new_protected:Npn \__xeCJK_long_punct_kerning:N #1
5054 {
5055   \dim_set:Nn \l__xeCJK_tmp_dim
5056   {
5057     \dim_max:n

```

```

5058         { \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim }
5059         { \c_zero_dim }
5060     }
5061     \l__xeCJK_save_punct_dim:nNNn { bound_width } #1 #1 { \l__xeCJK_tmp_dim }
5062     \dim_set:Nn \l__xeCJK_tmp_dim
5063     {
5064         \str_case:nnF {#1}
5065         {
5066             { ~~~~2014 }
5067             {
5068                 - \dim_max:nn
5069                 { \l__xeCJK_tmp_dim }
5070                 {
5071                     \dim_max:nn
5072                     {
5073                         \l__xeCJK_use_punct_dim:nN { dimen } #1 +
5074                         \l__xeCJK_use_punct_dim:nN { width } #1 -
5075                         \f@size pt - \f@size pt
5076                     }
5077                     {
5078                         \l__xeCJK_use_punct_dim:nN { width } #1 +
5079                         \l__xeCJK_use_punct_dim:nN { width } #1 -
5080                         \f@size pt - \f@size pt
5081                     }
5082                 }
5083             }
5084             { ~~~~2025 } { \c_zero_dim }
5085             { ~~~~2026 } { \c_zero_dim }
5086         }
5087         { -\l__xeCJK_tmp_dim }
5088     }
5089     \l__xeCJK_save_punct_dim:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
5090     \l__xeCJK_save_punct_skip:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
5091     \dim_add:Nn \l__xeCJK_tmp_dim
5092     { \dim_max:nn { \l__xeCJK_bound_dim } { \c_zero_dim } }
5093     \l__xeCJK_save_punct_dim:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
5094     \l__xeCJK_save_punct_skip:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
5095 }

```

\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 \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。

```

\__xeCJK_save_punct_kerning_plain:NN
\__xeCJK_save_punct_kerning:NN
\__xeCJK_save_left_right_kerning:NN
5096 \cs_new_protected:Npn \xeCJK_get_punct_kerning:NN #1#2
5097 {
5098     \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/kern/#1/#2 } }
5099     {
5100         \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
5101         { \__xeCJK_save_punct_kerning_plain:NN }
5102         { \__xeCJK_save_punct_kerning:NN }
5103         #1 #2
5104     }
5105 }
5106 \cs_new_protected:Npn \xeCJK_get_punct_kerning:oN
5107 { \exp_after:wN \xeCJK_get_punct_kerning:NN }
5108 \cs_new_protected:Npn \__xeCJK_save_punct_kerning_plain:NN #1#2
5109 {
5110     \l__xeCJK_save_punct_dim:nNNn { kern } #1 #2 { \c_zero_dim }
5111     \l__xeCJK_save_punct_dim:nNNn { bound_kern } #1 #2 { \c_zero_dim }
5112     \l__xeCJK_save_punct_dim:nNNn { bound_width } #1 #2 { \c_zero_dim }
5113     \l__xeCJK_save_punct_skip:nNNn { kern } #1 #2 { \c_zero_skip }
5114     \l__xeCJK_save_punct_skip:nNNn { bound_kern } #1 #2 { \c_zero_skip }
5115 }
5116 \cs_new_protected:Npn \__xeCJK_save_punct_kerning:NN #1#2
5117 {
5118     % 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。
5119     \UseInstance { xeCJK / punctuation } { \l__xeCJK_punct_style_tl }
5120     \__xeCJK_punct_if_right:NTF #1

```

```

5121     { \xeCJK_punct_kerning_process:NN #1 #2 }
5122     {
5123         \__xeCJK_punct_if_right:NTF #2
5124         { \__xeCJK_save_left_right_kerning:NN #1 #2 }
5125         { \xeCJK_punct_kerning_process:NN #1 #2 }
5126     }
5127 }
5128 \cs_new_protected:Npn \__xeCJK_save_left_right_kerning:NN #1#2
5129 {
5130     \bool_if:NTF \l__xeCJK_enabled_left_right_kerning_bool
5131     { \xeCJK_punct_kerning_process:NN #1 #2 }
5132     {
5133         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5134         {
5135             \tl_if_exist:cTF { g__xeCJK_punct/kern/#1/#2/tl }
5136             { \xeCJK_punct_kerning_process:NN #1 #2 }
5137             { \__xeCJK_save_punct_kerning_plain:NN #1 #2 }
5138         }
5139         { \__xeCJK_save_punct_kerning_plain:NN #1 #2 }
5140     }
5141 }

```

_xeCJK_punct_if_full_margin_dash:N 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。

```

5142 \prg_new_conditional:Npnn \__xeCJK_punct_if_full_margin_dash:N #1 { p , T , F , TF }
5143 {
5144     \str_if_eq:nnTF {#1} { ~~~~2014 }
5145     {
5146         \if_int_compare:w \xeCJK_token_value_class:N #1 =
5147             \xeCJK_class_num:n { PoZheHao }
5148         \prg_return_false:
5149     }
5150     \prg_return_true:
5151     \fi:
5152 }
5153 { \prg_return_false: }
5154 }

```

\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。

```

5155 \cs_new_protected:Npn \xeCJK_punct_margin_process:NN #1#2
5156 {
5157     \dim_set:Nn \l__xeCJK_tmp_dim
5158     {
5159         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5160         {
5161             \cs_if_exist_use:cF { g__xeCJK_punct_width/#2/tl }
5162             {
5163                 \tl_if_empty:NTF \g__xeCJK_punct_width_tl
5164                 { \__xeCJK_calc_punct_width:N #2 }
5165                 { \g__xeCJK_punct_width_tl }
5166             }
5167         }
5168         { \__xeCJK_calc_punct_width:N #2 }
5169     }
5170     \dim_set:Nn \l__xeCJK_margin_dim
5171     {
5172         \dim_max:nn
5173         { \l__xeCJK_margin_minimum_dim }
5174         {
5175             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
5176             {
5177                 \__xeCJK_punct_if_middle:NTF #2
5178                 {

```

```

5179         ( \l__xeCJK_tmp_dim
5180         - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5181         )
5182         \__xeCJK_punct_if_full_margin_dash:Nf #2 { / 2 }
5183     }
5184     {
5185         \bool_if:NTF \l__xeCJK_optimize_margin_bool
5186         {
5187             \dim_max:nn
5188             {
5189                 \dim_min:nn
5190                 { \l__xeCJK_bound_dim }
5191                 { \l__xeCJK_reverse_bound_dim }
5192             }
5193         }
5194         { \use:n }
5195         {
5196             \l__xeCJK_tmp_dim
5197             - \l__xeCJK_reverse_bound_dim
5198             - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5199         }
5200     }
5201 }
5202 {
5203     \bool_if:NTF \l__xeCJK_optimize_margin_bool
5204     { \dim_min:nn { \l__xeCJK_bound_dim } }
5205     { \use:n }
5206     { \__xeCJK_calc_margin_width:N #2 }
5207 }
5208 }
5209 }
5210 \__xeCJK_save_punct_dim:nNn { margin } #1 #2 { \l__xeCJK_margin_dim }
5211 }
5212 \dim_new:N \l__xeCJK_margin_dim
\__xeCJK_calc_punct_width:N 5213 \cs_new:Npn \__xeCJK_calc_punct_width:N #1
5214 {
5215     \__xeCJK_punct_if_middle:Nf #1
5216     { \__xeCJK_punct_width_or_ratio:nN { middle } }
5217     {
5218         \__xeCJK_punct_if_mixed_width:Nf #1
5219         { \__xeCJK_punct_width_or_ratio:nN { mixed } }
5220         { \__xeCJK_punct_width_or_ratio:nN { fixed } }
5221     }
5222     #1
5223 }
\__xeCJK_calc_margin_width:N 5224 \cs_new:Npn \__xeCJK_calc_margin_width:N #1
5225 {
5226     \__xeCJK_punct_if_middle:Nf #1
5227     {
5228         \dim_compare:nNf \l__xeCJK_middle_margin_width_dim < \c_max_dim
5229         { \l__xeCJK_middle_margin_width_dim }
5230         {
5231             \__xeCJK_dim_ratio:Nn \l__xeCJK_middle_margin_ratio_fp
5232             { ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / 2 }
5233         }
5234     }
5235     {
5236         \__xeCJK_punct_if_mixed_width:Nf #1
5237         { \__xeCJK_margin_width_or_ratio:n { mixed } }
5238         { \__xeCJK_margin_width_or_ratio:n { fixed } }
5239     }
5240 }
\__xeCJK_dim_ratio:Nn 5241 \cs_new:Npn \__xeCJK_dim_ratio:Nn #1#2
5242 { \fp_to_dim:n { #1 \dim_to_fp:n {#2} } }
5243 \cs_generate_variant:Nn \__xeCJK_dim_ratio:Nn { c }

```

```

\xCJK_punct_offset_process:NN 5244 \cs_new_protected:Npn \xeCJK_punct_offset_process:NN #1#2
5245 {
5246   \dim_set:Nn \l__xeCJK_tmp_dim
5247   {
5248     \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5249     {
5250       \cs_if_exist_use:cF { g__xeCJK_punct_bound_width/#2/tl }
5251       {
5252         \tl_if_empty:NTF \g__xeCJK_punct_bound_width_tl
5253         { \__xeCJK_punct_width_or_ratio:nN { bound } #2 }
5254         { \g__xeCJK_punct_bound_width_tl }
5255       }
5256     }
5257     { \__xeCJK_punct_width_or_ratio:nN { bound } #2 }
5258   }
5259   \dim_set:Nn \l__xeCJK_tmp_dim
5260   {
5261     \bool_if:NTF \l__xeCJK_enabled_hanging_bool
5262     { \use:n }
5263     { \dim_max:nn { \l__xeCJK_margin_minimum_dim } }
5264     {
5265       \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
5266       {
5267         \__xeCJK_punct_if_middle:NTF #2
5268         {
5269           \l__xeCJK_tmp_dim
5270           - \l__xeCJK_margin_dim
5271           - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5272         }
5273         {
5274           \l__xeCJK_tmp_dim
5275           - \l__xeCJK_reverse_bound_dim
5276           - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5277         }
5278       }
5279       {
5280         \bool_if:NTF \l__xeCJK_optimize_margin_bool
5281         { \dim_min:nn { \l__xeCJK_bound_dim } }
5282         { \use:n }
5283         { \__xeCJK_margin_width_or_ratio:n { bound } }
5284       }
5285     }
5286   }
5287   \__xeCJK_save_punct_dim:nNNn { offset } #1 #2
5288   { \l__xeCJK_tmp_dim }
5289   \__xeCJK_save_punct_dim:nNNn { rule } #1 #2
5290   { \l__xeCJK_tmp_dim - \l__xeCJK_bound_dim }
5291   \__xeCJK_save_punct_dim:nNNn { rule } \l__xeCJK_reverse_tl #2
5292   { \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim }
5293   \__xeCJK_save_punct_dim:nNNn { glue } #1 #2
5294   { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
5295   \__xeCJK_save_punct_skip:nNNnn { glue } #1 #2
5296   { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
5297   {
5298     \__xeCJK_punct_if_middle:NTF #2
5299     {
5300       ( \__xeCJK_use_punct_dim:nN { width } #2 -
5301         \__xeCJK_use_punct_dim:nN { dimen } #2 )
5302       \__xeCJK_punct_if_full_margin_dash:NF #2 { / 2 }
5303       - \l__xeCJK_margin_dim
5304     }
5305     { \l__xeCJK_bound_dim - \l__xeCJK_margin_dim }
5306   }
5307   {
5308     \__xeCJK_punct_if_middle:NTF #2
5309     { \l__xeCJK_margin_dim / 2 }
5310     { \l__xeCJK_margin_dim - \l__xeCJK_reverse_bound_dim }

```

```
5311     }
5312 }
```

```
\_xeCJK_punct_width_or_ratio:nN 5313 \cs_new:Npn \_xeCJK_punct_width_or_ratio:nN #1#2
5314 {
5315     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
5316     { \use:c { l__xeCJK_#1_punct_width_dim } }
5317     {
5318         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_punct_ratio_fp } }
5319         { \c_max_dim }
5320         {
5321             \_xeCJK_dim_ratio:cn
5322             { l__xeCJK_#1_punct_ratio_fp }
5323             { \_xeCJK_use_punct_dim:nN { width } #2 }
5324         }
5325     }
5326 }
```

```
\_xeCJK_margin_width_or_ratio:n 5327 \cs_new:Npn \_xeCJK_margin_width_or_ratio:n #1
5328 {
5329     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
5330     { \use:c { l__xeCJK_#1_margin_width_dim } }
5331     {
5332         \_xeCJK_dim_ratio:cn
5333         { l__xeCJK_#1_margin_ratio_fp }
5334         { \l__xeCJK_bound_dim }
5335     }
5336     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
5337     { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
5338 }
```

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

```
5339 \cs_new_protected:Npn \xeCJK_punct_kerning_process:NN #1#2
5340 {
5341     \dim_set:Nn \l__xeCJK_margin_dim
5342     { \_xeCJK_original_kerning_margin:NN #1 #2 }
5343     \dim_set:Nn \l__xeCJK_minimum_bound_dim
5344     { \_xeCJK_punct_min_bound:NN #1 #2 }
5345     \_xeCJK_punct_if_long:NTF #1
5346     { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
5347     {
5348         \_xeCJK_punct_if_long:NT #2
5349         { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
5350     }
5351     \dim_set:Nn \l__xeCJK_kerning_margin_dim
5352     {
5353         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5354         {
5355             \cs_if_exist_use:cF { g__xeCJK_punct/kern/#1/#2/t1 }
5356             { \_xeCJK_punct_kerning_process_aux:NN #1 #2 }
5357         }
5358         { \_xeCJK_punct_kerning_process_aux:NN #1 #2 }
5359     }
5360     \_xeCJK_save_kerning:nnNN { kern } { bound } #1 #2
5361     \_xeCJK_save_punct_dim:nnNN { bound_width } #1 #2
5362     { \l__xeCJK_kerning_margin_dim - \l__xeCJK_tmp_dim }
5363     \_xeCJK_punct_if_right:NTF #1
5364     {
5365         \_xeCJK_punct_if_right:NTF #2
5366         { \_xeCJK_save_kerning:nnnNN { bound_kern } { offset } { bound } }
5367         { \_xeCJK_save_kerning:nnNN { bound_kern } { offset } }
5368     }
5369     {
5370         \_xeCJK_punct_if_right:NTF #2
5371         { \_xeCJK_save_kerning:nnNN { bound_kern } { bound } }
5372         { \_xeCJK_save_kerning:nnnNN { bound_kern } { bound } { offset } }
```

```

5373     }
5374     #1 #2
5375 }
5376 \cs_new:Npn \__xeCJK_punct_kerning_process_aux:NN #1#2
5377 {
5378     \bool_if:NTF \l__xeCJK_enabled_kerning_bool
5379     { \__xeCJK_calc_kerning_margin:NN #1 #2 }
5380     { \l__xeCJK_margin_dim }
5381 }
5382 \dim_new:N \l__xeCJK_minimum_bound_dim
5383 \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。

```

5384 \cs_new_protected:Npn \__xeCJK_save_kerning:nnNN #1#2
5385 { \__xeCJK_save_kerning:nnnNN {#1} {#2} {#2} }
5386 \cs_new_protected:Npn \__xeCJK_save_kerning:nnnNN #1#2#3#4#5
5387 {
5388     \dim_set:Nn \l__xeCJK_tmp_dim
5389     {
5390         \l__xeCJK_kerning_margin_dim
5391         - ( \__xeCJK_use_punct_dim:nNN {#2} \c__xeCJK_right_tl #4 )
5392         - ( \__xeCJK_use_punct_dim:nNN {#3} \c__xeCJK_left_tl #5 )
5393     }
5394     \__xeCJK_save_punct_dim:nNn {#1} #4 #5 { \l__xeCJK_tmp_dim }
5395     \__xeCJK_save_punct_skip:nNnnn {#1} #4 #5
5396     { \l__xeCJK_tmp_dim }
5397     { \l__xeCJK_margin_dim - \l__xeCJK_kerning_margin_dim }
5398     { \l__xeCJK_kerning_margin_dim - \l__xeCJK_minimum_bound_dim }
5399 }

```

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

```

5400 \cs_new:Npn \__xeCJK_original_kerning_margin:NN #1#2
5401 {
5402     \dim_eval:n
5403     {
5404         \__xeCJK_use_punct_dim:nNN
5405         { \__xeCJK_punct_if_right:NTF #1 { margin } { bound } } \c__xeCJK_right_tl #1
5406         +
5407         \__xeCJK_use_punct_dim:nNN
5408         { \__xeCJK_punct_if_right:NTF #2 { bound } { margin } } \c__xeCJK_left_tl #2
5409     }
5410 }

```

```

\__xeCJK_calc_kerning_margin:NN 5411 \cs_new:Npn \__xeCJK_calc_kerning_margin:NN #1#2
\__xeCJK_calc_kerning_margin_aux:NN 5412 {
5413     \dim_max:nn
5414     { \l__xeCJK_kerning_margin_minimum_dim }
5415     {
5416         \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
5417         { \l__xeCJK_minimum_bound_dim }
5418         {
5419             \bool_if:NTF \l__xeCJK_optimize_kerning_bool
5420             { \dim_max:nn { \l__xeCJK_minimum_bound_dim } }
5421             { \use:n }
5422             { \__xeCJK_calc_kerning_margin_aux:NN #1 #2 }
5423         }
5424     }
5425 }
5426 \cs_new:Npn \__xeCJK_calc_kerning_margin_aux:NN #1#2
5427 {
5428     \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
5429     { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
5430     {
5431         \fp_if_nan:nTF { \l__xeCJK_kerning_total_ratio_fp }
5432         {
5433             \xeCJK_if_same_class:NNTF #1 #2

```

```

5434         { \_xeCJK_kerning_width_or_ratio:nNN { same } }
5435         { \_xeCJK_kerning_width_or_ratio:nNN { different } }
5436     }
5437     {
5438         \_xeCJK_calc_kerning_margin:nNN
5439         {
5440             \_xeCJK_dim_ratio:Nn \l__xeCJK_kerning_total_ratio_fp
5441             {
5442                 \_xeCJK_use_punct_dim:nN { width } #1 +
5443                 \_xeCJK_use_punct_dim:nN { width } #2
5444             }
5445         }
5446     }
5447 }
5448 #1 #2
5449 }

\_xeCJK_kerning_width_or_ratio:nNN 5450 \cs_new:Npn \_xeCJK_kerning_width_or_ratio:nNN #1#2#3
5451 {
5452     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
5453     { \use:c { l__xeCJK_#1_align_margin_dim } }
5454     {
5455         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_align_ratio_fp } }
5456         {
5457             \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
5458             { \l__xeCJK_kerning_margin_width_dim \use:none:n }
5459             { \_xeCJK_dim_ratio:Nn \l__xeCJK_kerning_ratio_fp }
5460         }
5461         { \_xeCJK_dim_ratio:cn { l__xeCJK_#1_align_ratio_fp } }
5462         { \l__xeCJK_margin_dim }
5463     }
5464 }

\_xeCJK_punct_min_bound:NN 5465 \cs_new:Npn \_xeCJK_punct_min_bound:NN #1#2
5466 {
5467     \dim_max:nn
5468     {
5469         \dim_min:nn
5470         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 }
5471         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 }
5472     }
5473     {
5474         \dim_min:nn
5475         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #2 }
5476         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #2 }
5477     }
5478 }

```

#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。

```

5479 \cs_new:Npn \_xeCJK_calc_kerning_margin:nNN #1#2#3
5480 {
5481     \dim_eval:n
5482     {
5483         (#1)
5484         - ( \_xeCJK_use_punct_dim:nNN
5485             { \_xeCJK_punct_if_right:NTF #2 { bound } { margin } }
5486             \c__xeCJK_left_tl #2 )
5487         - ( \_xeCJK_use_punct_dim:nNN
5488             { \_xeCJK_punct_if_right:NTF #3 { margin } { bound } }
5489             \c__xeCJK_right_tl #3 )
5490         - ( \_xeCJK_use_punct_dim:nN { dimen } #2 )
5491         - ( \_xeCJK_use_punct_dim:nN { dimen } #3 )
5492     }
5493 }

```

计算标点的左右实际边距和实际尺寸。

```

5494 \cs_new_protected:Npn \xeCJK_calc_punct_dimen:N #1

```



```

5495 {
5496   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl #1
5497   { \xeCJK_glyph_bounds:NN 1 #1 }
5498   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl #1
5499   { \xeCJK_glyph_bounds:NN 3 #1 }
5500   \__xeCJK_save_punct_dim:nNn { width } #1
5501   { \tex_fontcharwd:D \tex_font:D `#1 }
5502   \__xeCJK_save_punct_dim:nNn { dimen } #1
5503   {
5504     ( \__xeCJK_use_punct_dim:nN { width } #1 ) -
5505     ( \__xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 ) -
5506     ( \__xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 )
5507   }
5508 }

```

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

```

5509 \cs_new:Npn \xeCJK_glyph_bounds:NN #1#2
5510 { \tex_XeTeXglyphbounds:D #1 ~ \tex_XeTeXcharglyph:D `#2 \exp_stop_f: }

```

```

PunctStyle 5511 \keys_define:nn { xeCJK / options }
5512 { PunctStyle .code:n = \exp_args:Ne \__xeCJK_set_punct_style:n {#1} }
5513 \cs_new_protected:Npn \__xeCJK_set_punct_style:n #1
5514 {
5515   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5516   { \tl_set:Nn \l_xeCJK_punct_style_tl {#1} }
5517   {
5518     \prop_get:NnNF \c__xeCJK_punct_style_alias_prop
5519     {#1} \l_xeCJK_punct_style_tl
5520     { \__xeCJK_error:ne { punct-style-unknown } {#1} }
5521   }
5522 }
5523 \prop_const_from_keyval:Nn \c__xeCJK_punct_style_alias_prop
5524 {
5525   halfwidth      = banjiao ,
5526   fullwidth      = quanjiao ,
5527   mixedwidth     = kaiming ,
5528   marginkerning  = hangmobanjiao ,
5529   plain          = plain
5530 }
5531 \tl_new:N \l_xeCJK_punct_style_tl
5532 \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
5533 \__xeCJK_msg_new:nn { punct-style-unknown }
5534 {
5535   Punctuation~style~"#1"~is~unknown. \\\
5536   The~available~styles~are~listed~as~follow.\\
5537   "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq
5538   { ~and~ } { ,~ } { ,~and~ }".\\
5539 }

```

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

```

5540 \cs_new_protected:Npn \__xeCJK_trim_spaces:n #1
5541 {
5542   \tl_set:Ne \ProcessedArgument
5543   { \exp_args:Ne \tl_trim_spaces:n {#1} }
5544 }

```

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

```

5545 \NewDocumentCommand \xeCJKDeclarePunctStyle
5546 { > { \__xeCJK_trim_spaces:n } m m }
5547 {
5548   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5549   { \__xeCJK_warning:ne { punct-style-already-defined } {#1} }
5550   { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
5551   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
5552 }

```

```

5553 \seq_new:N \g__xeCJK_punct_style_seq
5554 \__xeCJK_msg_new:nn { punct-style-already-defined }
5555 {
5556   Punctuation~style~"#1"~is~already~defined!. \\\
5557   The~existing~style~of~"#1"~will~be~overwritten.\\
5558 }
5559 \@onlypreamble \xeCJKDeclarePunctStyle

```

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

```

5560 \NewDocumentCommand \xeCJKEditPunctStyle
5561 { > { \__xeCJK_trim_spaces:n } m m }
5562 {
5563   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5564   { \EditInstance { xeCJK / punctuation } {#1} {#2} }
5565   { \__xeCJK_error:ne { punct-style-unknown } {#1} }
5566 }
5567 \@onlypreamble \xeCJKEditPunctStyle

```

默认设置即为全角格式。

```

5568 \xeCJKDeclarePunctStyle { quanjiao }
5569 {
5570   optimize-kerning = true ,
5571   enabled-left-right-kerning = false
5572 }

5573 \xeCJKDeclarePunctStyle { hangmobanjiang } { enabled-kerning = false }

5574 \xeCJKDeclarePunctStyle { banjiao }
5575 {
5576   fixed-punct-ratio = 0.5 ,
5577   optimize-margin = true ,
5578   kerning-total-ratio = 0.5 ,
5579   optimize-kerning = true
5580 }

5581 \xeCJKDeclarePunctStyle { kaiming }
5582 {
5583   fixed-punct-ratio = 0.5 ,
5584   mixed-punct-ratio = 1.0 ,
5585   optimize-margin = true ,
5586   kerning-total-ratio = 0.5 ,
5587   optimize-kerning = true
5588 }

5589 \xeCJKDeclarePunctStyle { CCT }
5590 {
5591   fixed-punct-ratio = 0.7 ,
5592   optimize-margin = true ,
5593   kerning-total-ratio = 0.6 ,
5594   optimize-kerning = true
5595 }

```

5.11 后备字体

`AutoFallBack` 后备字体的宏包选项声明。

```

5596 \keys_define:nn { xeCJK / options }
5597 {
5598   AutoFallBack .choice: ,
5599   AutoFallBack / true .code:n =
5600   {
5601     \cs_set_eq:NN \xeCJK_fallback_symbol:NN
5602     \__xeCJK_fallback_symbol:NN
5603     \cs_set_eq:NN \xeCJK_fallback_punct_symbol:NN
5604     \__xeCJK_fallback_punct_symbol:NN
5605     \cs_set_eq:NN \xeCJK_clear_fallback_font:
5606     \__xeCJK_clear_fallback_font:

```

```

5607     } ,
5608     AutoFallBack / false .code:n =
5609     {
5610         \xeCJK_cs_clear:N \xeCJK_fallback_symbol:NN
5611         \xeCJK_cs_clear:N \xeCJK_fallback_punct_symbol:NN
5612         \xeCJK_cs_clear:N \xeCJK_clear_fallback_font:
5613     } ,
5614     AutoFallBack .default:n = { true } ,
5615     fallback .meta:n = { AutoFallBack = true }
5616 }

```

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
5617 \cs_new_protected:Npn \__xeCJK_fallback_symbol:NN #1#2
5618 {
5619     \xeCJK_reset_fallback_font:
5620     \xeCJK_glyph_if_exist:NF #2
5621     { \__xeCJK_fallback_symbol_aux:NN }
5622     #1#2
5623 }
5624 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol:NN #1#2
5625 {
5626     \xeCJK_reset_fallback_font:
5627     \xeCJK_glyph_if_exist:NF #2
5628     { \__xeCJK_fallback_punct_symbol_aux:NN }
5629     #1#2
5630 }
5631 \cs_new_eq:NN \xeCJK_fallback_symbol:NN \prg_do_nothing:
5632 \cs_new_eq:NN \xeCJK_fallback_punct_symbol:NN \prg_do_nothing:
5633 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:NN
5634 {
5635     \__xeCJK_fallback_symbol_aux:nnNN
5636     { \CJK@family }
5637     { \l_xeCJK_family_tl }
5638 }
5639 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol_aux:NN
5640 {
5641     \__xeCJK_fallback_symbol_aux:nnNN
5642     { \CJK@punctfamily }
5643     { \l_xeCJK_punct_family_tl }
5644 }
5645 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:nnNN
5646 {
5647     \cs_set_protected:Npe \xeCJK_reset_fallback_font:
5648     {
5649         \tex_the:D \tex_font:D
5650         \xeCJK_clear_fallback_font:
5651     }
5652     \exp_args:Nee \__xeCJK_fallback_loop:nnNN
5653 }
5654 \cs_new_protected:Npn \__xeCJK_clear_fallback_font:
5655 { \cs_set_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing: }
5656 \cs_new_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing:
5657 \cs_new_eq:NN \xeCJK_clear_fallback_font: \prg_do_nothing:

```

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。

```

5658 \cs_new_protected:Npn \__xeCJK_fallback_loop:nnNN
5659 {
5660     \cs_set_eq:NN \__xeCJK_fallback_loop:TF \use_i:nn
5661     \__xeCJK_fallback_loop:nnnNN { FallBack }
5662 }
5663 \cs_new_protected:Npn \__xeCJK_fallback_loop:nnnNN #1#2#3
5664 {
5665     \xeCJK_select_fallback_font:nnn {#1} {#2} {#3}
5666     \__xeCJK_fallback_loop:TF

```

```

5667     { \_xeCJK_fallback_loop_aux:nnnNN }
5668     { \_xeCJK_fallback_missing_glyph:nnnNN }
5669     {#1} {#2} {#3}
5670 }
5671 \cs_new_protected:Npn \_xeCJK_fallback_loop_aux:nnnNN #1#2#3#4#5
5672 {
5673     \xeCJK_glyph_if_exist:NF #5
5674     { \_xeCJK_fallback_loop:nnnNN { #1/FallBack } {#2} {#3} }
5675     #4#5
5676 }
5677 \cs_new_protected:Npn \_xeCJK_fallback_missing_glyph:nnnNN #1#2#3#4#5
5678 {
5679     \_xeCJK_warning:neee { missing-glyph } {#1} {#2} {#5}
5680     #4#5
5681 }
5682 \cs_new_protected:Npn \xeCJK_select_fallback_font:nnn #1#2
5683 {
5684     \_xeCJK_select_fallback_font:cnnn
5685     { \_xeCJK_font_csname:n { #2/#1 } } {#1} {#2}
5686 }
5687 \cs_new_protected:Npn \_xeCJK_select_fallback_font:Nnnn #1
5688 {
5689     \cs_if_exist:NF #1
5690     { \_xeCJK_fallback_font_initial:NNnnn }
5691     #1 \use_none:nnn
5692 }
5693 \cs_generate_variant:Nn \_xeCJK_select_fallback_font:Nnnn { c }
5694 \cs_new_protected:Npn \_xeCJK_fallback_font_initial:NNnnn #1#2#3#4#5
5695 {
5696     \xeCJK_family_if_exist:nTF { #5/#3 }
5697     { \_xeCJK_font_initial:Nn #1 { #5/#3 } }
5698     { \_xeCJK_fallback_font_initial_auxi:Nnnn #1 {#5} {#3} {#4} }
5699     #1
5700 }
5701 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxi:Nnnn #1
5702 {
5703     \exp_args:NNe \_xeCJK_fallback_font_initial_auxii:Nnnnn
5704     #1 { \CJKfamilydefault }
5705 }
5706 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxii:Nnnnn #1#2#3
5707 {
5708     \str_if_eq:nnTF {#2} {#3}
5709     { \_xeCJK_fallback_loop_end:Nnnn }
5710     { \_xeCJK_fallback_font_initial_auxiii:Nnnn }
5711     #1 {#2}
5712 }
5713 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxiii:Nnnn #1#2
5714 {
5715     \xeCJK_family_if_exist:nTF {#2}
5716     { \_xeCJK_fallback_font_initial_auxiv:Nnnn }
5717     { \_xeCJK_fallback_loop_end:Nnnn }
5718     #1 {#2}
5719 }
5720 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxiv:Nnnn #1#2#3#4
5721 {
5722     \_xeCJK_font_initial:Nn #1 {#2}
5723     \exp_args:Nc \_xeCJK_fallback_font_initial_auxiii:Nnnn
5724     { \_xeCJK_font_csname:n { #4/#3/FallBack } }
5725     { #2/FallBack } { #3/FallBack } {#4}
5726 }
5727 \cs_new_eq:NN \_xeCJK_fallback_loop:TF \use_i:nn
5728 \cs_new_protected:Npn \_xeCJK_fallback_loop_end:Nnnn #1#2#3#4
5729 { \cs_gset_eq:NN #1 \_xeCJK_fallback_loop_end: }
5730 \cs_new_protected:Npn \_xeCJK_fallback_loop_end:
5731 { \cs_set_eq:NN \_xeCJK_fallback_loop:TF \use_ii:nn }
5732 \_xeCJK_msg_new:nn { missing-glyph }
5733 {

```

```

5734 CJKfamily~\__xeCJK_msg_family_map:n {#2}'~(#{1}~
5735 does~not~contain~glyph~`#3'~
5736 ( U + \int_to_Hex:n { `#3 } )~\msg_line_context:.
5737 }

\setCJKfallbackfamilyfont 5738 \NewDocumentCommand \setCJKfallbackfamilyfont { m o m }
5739 {
5740   \__xeCJK_pass_args:nnnn
5741   { \xeCJK_set_family_fallback:nnn {#1} } {#2} {#3}
5742   { }
5743 }

xeCJK_set_family_fallback:nnn 5744 \cs_new_protected:Npn \xeCJK_set_family_fallback:nnn #1#2#3
5745 {
5746   \group_begin:
5747   \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
5748   \prop_get:NoNF \g__xeCJK_family_font_name_prop
5749   \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
5750   { \tl_clear:N \l__xeCJK_font_name_tl }
5751   \clist_set:Nn \l__xeCJK_public_options_clist {#2}
5752   \clist_map_function:nN {#3} \__xeCJK_set_family_fallback:n
5753   \group_end:
5754 }
5755 \cs_new_protected:Npn \__xeCJK_set_family_fallback:n #1
5756 {
5757   \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /Fallback }
5758   \__xeCJK_get_sub_features:on \l__xeCJK_fallback_family_tl {#1}
5759   \clist_concat:NNN \l__xeCJK_sub_font_options_clist
5760   \l__xeCJK_public_options_clist
5761   \l__xeCJK_sub_font_options_clist
5762   \exp_args:Nooo \xeCJK_set_family:nnn
5763   \l__xeCJK_fallback_family_tl
5764   \l__xeCJK_sub_font_options_clist
5765   \l__xeCJK_sub_font_name_tl
5766 }
5767 \tl_new:N \l__xeCJK_fallback_family_tl
5768 \clist_new:N \l__xeCJK_public_options_clist

```

5.12 CJK 字体族声明方式

```

5769 \bool_new:N \g__xeCJK_auto_fake_bold_bool
5770 \bool_new:N \g__xeCJK_auto_fake_slant_bool
5771 \fp_new:N \g__xeCJK_embolden_factor_fp
5772 \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。

```

5773 \keys_define:nn { xeCJK / options }
5774 {
5775   AutoFakeBold .choices:nn = { true , false }
5776   { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_bold_bool } ,
5777   AutoFakeBold / unknown .code:n =
5778   {
5779     \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
5780     \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
5781   } ,
5782   AutoFakeBold .default:n = { true } ,
5783   AutoFakeSlant .choices:nn = { true , false }
5784   { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_slant_bool } ,
5785   AutoFakeSlant / unknown .code:n =
5786   {
5787     \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
5788     \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
5789   } ,
5790   AutoFakeSlant .default:n = { true } ,
5791   EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
5792   SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
5793   BoldFont .meta:n = { AutoFakeBold = true } ,

```

```

5794     boldfont .meta:n = { AutoFakeBold = true } ,
5795     SlantFont .meta:n = { AutoFakeSlant = true } ,
5796     slantfont .meta:n = { AutoFakeSlant = true }
5797 }

```

\xeCJK_new_sub_key:n 用于定义 CJK 子区字体和备用字体的选项。

\g__xeCJK_sub_key_seq

```

5798 \seq_new:N \g__xeCJK_sub_key_seq
5799 \cs_new_protected:Npn \xeCJK_new_sub_key:n #1
5800 {
5801     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
5802     \keys_define:nn { xeCJK / features }
5803     {
5804         #1 .code:n =
5805         {
5806             \tl_if_blank:nTF {##1}
5807             {
5808                 \prop_clear:N \l__xeCJK_sub_key_prop
5809                 \tl_set:Nx \l__xeCJK_sub_family_name_tl
5810                 { \l__xeCJK_family_name_tl /#1 }
5811                 \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
5812             }
5813             {
5814                 \tl_clear:N \l__xeCJK_sub_family_name_tl
5815                 \str_if_eq:nnTF {##1} { * }
5816                 { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
5817                 { \__xeCJK_get_sub_features:nn {#1} {##1} }
5818             }
5819         } ,
5820         #1 .default:n = { }
5821     }
5822 }

```

__xeCJK_get_sub_features:nn

__xeCJK_get_sub_features:w

```

5823 \cs_new_protected:Npn \__xeCJK_get_sub_features:nn #1#2
5824 {
5825     \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
5826     \clist_clear:N \l__xeCJK_sub_font_options_clist
5827     \exp_after:wN \__xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
5828     \q_mark [ \q_nil ] \q_mark \q_stop
5829     \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
5830     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
5831     { \tl_replace_all:Nno \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
5832     \prop_put:Nne \l__xeCJK_sub_key_prop {#1}
5833     {
5834         { \exp_not:o \l__xeCJK_sub_font_options_clist }
5835         { \exp_not:o \l__xeCJK_sub_font_name_tl }
5836     }
5837 }
5838 \cs_new_protected:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
5839 {
5840     \quark_if_nil:nTF {#2}
5841     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
5842     {
5843         \tl_set:Nx \l__xeCJK_sub_font_name_tl
5844         { \xeCJK_tl_remove_outer_braces:n {#3} }
5845         \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
5846         { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
5847         { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
5848     }
5849 }
5850 \tl_new:N \l__xeCJK_sub_family_name_tl
5851 \tl_new:N \l__xeCJK_sub_font_name_tl
5852 \clist_new:N \l__xeCJK_sub_font_options_clist
5853 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { o }
5854 \cs_generate_variant:Nn \tl_replace_all:Nnn { Nno }

```

Fallback 5855 \xeCJK_new_sub_key:n { Fallback }

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

```
5856 \keys_define:nn { xeCJK / features }
5857 {
5858   BoldFont .tl_set:N = \l__xeCJK_font_name_bf_tl ,
5859   ItalicFont .tl_set:N = \l__xeCJK_font_name_it_tl
5860 }
```

```
AutoFakeBold 5861 \keys_define:nn { xeCJK / features }
AutoFakeSlant 5862 {
5863   AutoFakeBold .choice: ,
5864   AutoFakeBold / true .code:n =
5865   {
5866     \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
5867     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
5868   } ,
5869   AutoFakeBold / false .code:n =
5870   { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
5871   AutoFakeBold / unknown .code:n =
5872   {
5873     \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
5874     \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
5875   } ,
5876   AutoFakeBold .default:n = { true } ,
5877   AutoFakeSlant .choice: ,
5878   AutoFakeSlant / true .code:n =
5879   {
5880     \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
5881     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
5882   } ,
5883   AutoFakeSlant / false .code:n =
5884   { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
5885   AutoFakeSlant / unknown .code:n =
5886   {
5887     \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
5888     \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
5889   } ,
5890   AutoFakeSlant .default:n = { true }
5891 }
```

```
\__xeCJK_set_family_initial: 5892 \cs_new_protected:Npn \__xeCJK_set_family_initial:
5893 {
5894   \int_gincr:N \g__xeCJK_family_int
5895   \prop_clear:N \l__xeCJK_sub_key_prop
5896   \tl_clear:N \l__xeCJK_font_name_bf_tl
5897   \tl_clear:N \l__xeCJK_font_name_it_tl
5898   \tl_clear:N \l__xeCJK_sub_family_name_tl
5899   \clist_clear:N \l__xeCJK_fontspec_options_clist
5900   \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
5901   \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
5902   \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
5903   \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
5904 }
5905 \int_new:N \g__xeCJK_family_int
5906 \prop_new:N \l__xeCJK_sub_key_prop
5907 \clist_new:N \l__xeCJK_fontspec_options_clist
5908 \bool_new:N \l__xeCJK_auto_fake_bold_bool
5909 \bool_new:N \l__xeCJK_auto_fake_slant_bool
5910 \fp_new:N \l__xeCJK_embolden_factor_fp
5911 \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。

```
5912 \cs_new_protected:Npn \xeCJK_set_family:nnn #1#2#3
5913 {
5914   \group_begin:
5915   \__xeCJK_set_family_initial:
5916   \tl_set:Nn \l__xeCJK_family_name_tl {#1}
```

```

5917 \clist_set:Nn \l__xeCJK_font_options_clist {#2}
5918 \tl_set:Nn \l__xeCJK_font_name_tl {#3}
5919 \clist_concat:NNN \l__xeCJK_font_options_clist
5920 \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
5921 \keys_set_known:noN { xeCJK / features }
5922 \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
5923 \__xeCJK_binding_sub_family:
5924 \__xeCJK_parse_font_shape:
5925 \__xeCJK_check_family:o \l__xeCJK_family_name_tl
5926 \__xeCJK_gset_family_cs:n { \l__xeCJK_family_name_tl }
5927 \__xeCJK_save_family_info:
5928 \__xeCJK_set_sub_block_family:
5929 \group_end:
5930 }
5931 \tl_new:N \l__xeCJK_family_name_tl
5932 \tl_new:N \l__xeCJK_font_name_tl
5933 \clist_new:N \l__xeCJK_font_options_clist
5934 \cs_generate_variant:Nn \xeCJK_set_family:nnn { e , o }

```

```

\__xeCJK_binding_sub_family: 5935 \cs_new_protected:Npn \__xeCJK_binding_sub_family:
5936 {
5937   \tl_if_empty:NF \l__xeCJK_sub_family_name_tl
5938   { \tl_set_eq:NN \l__xeCJK_family_name_tl \l__xeCJK_sub_family_name_tl }
5939 }

```

```

\__xeCJK_gset_family_cs:n 5940 \cs_new_protected:Npn \__xeCJK_gset_family_cs:n #1
5941 {
5942   \cs_gset_protected:cpe { \__xeCJK_family_csname:n {#1} }
5943   {
5944     \group_begin:
5945     \exp_not:n { \cs_set_eq:NN \xeCJK@fontfamily \use_none:n }
5946     \exp_not:n { \fontspec_gset_family:Nnn \g__xeCJK_fontspec_family_tl }
5947     { \exp_not:o \l__xeCJK_fontspec_options_clist }
5948     { \exp_not:o \l__xeCJK_font_name_tl }
5949     \__xeCJK_gset_family_nfss_cs:no
5950     {#1} { \exp_not:N \g__xeCJK_fontspec_family_tl }
5951     \group_end:
5952     \tl_set_eq:NN \exp_not:N \l__xeCJK_fontspec_family_tl
5953     \exp_not:N \g__xeCJK_fontspec_family_tl
5954   }
5955 }
5956 \tl_new:N \g__xeCJK_fontspec_family_tl
5957 \tl_new:N \l__xeCJK_fontspec_family_tl

```

```

\__xeCJK_check_family:n 5958 \cs_new_protected:Npn \__xeCJK_check_family:n #1
5959 {
5960   \prop_gpop:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
5961   {
5962     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
5963     {
5964       \cs_undefine:c { \__xeCJK_family_csname:n {#1} }
5965       \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} }
5966     }
5967     \__xeCJK_warning:nee { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
5968   }
5969 }
5970 \cs_generate_variant:Nn \__xeCJK_check_family:n { o }
5971 \__xeCJK_msg_new:nn { CJKfamily-redef }
5972 { Redefining~CJKfamily~\__xeCJK_msg_family_map:n {#1}'~(#{2}). }

```

```

\__xeCJK_parse_font_shape: 5973 \cs_new_protected:Npn \__xeCJK_parse_font_shape:
5974 {
5975   \tl_if_blank:oTF \l__xeCJK_font_name_bf_tl
5976   {
5977     \bool_if:NT \l__xeCJK_auto_fake_bold_bool
5978     {
5979       \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5980       { AutoFakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }

```



```

5981     }
5982   }
5983   {
5984     \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5985     { BoldFont = { \exp_not:o \l__xeCJK_font_name_bf_tl } }
5986   }
5987   \tl_if_blank:oTF \l__xeCJK_font_name_it_tl
5988   {
5989     \bool_if:NT \l__xeCJK_auto_fake_slant_bool
5990     {
5991       \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5992       { AutoFakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
5993     }
5994   }
5995   {
5996     \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5997     { ItalicFont = { \exp_not:o \l__xeCJK_font_name_it_tl } }
5998   }
5999 }

```

```

\g__xeCJK_family_name_prop 6000 \prop_new:N \g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop 6001 \prop_new:N \g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop 6002 \prop_new:N \g__xeCJK_family_font_options_prop

\__xeCJK_save_family_info: 6003 \cs_new_protected:Npn \__xeCJK_save_family_info:
6004 {
6005   \exp_args:Nooo \__xeCJK_save_family_info:nnn
6006   \l__xeCJK_family_name_tl
6007   \l__xeCJK_font_name_tl
6008   \l__xeCJK_font_options_clist
6009 }
6010 \cs_new_protected:Npn \__xeCJK_save_family_info:nnn #1#2#3
6011 {
6012   \prop_gput:Nnn \g__xeCJK_family_font_name_prop {#1} {#2}
6013   \prop_gput:Nnn \g__xeCJK_family_font_options_prop {#1} {#3}
6014 }

\__xeCJK_set_sub_block_family: 6015 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:
6016 {
6017   \prop_if_empty:NF \l__xeCJK_sub_key_prop
6018   {
6019     \prop_map_function:NN
6020     \l__xeCJK_sub_key_prop
6021     \__xeCJK_set_sub_block_family:nn
6022   }
6023 }
6024 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:nn #1#2
6025 {
6026   \tl_set:Ne \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/#1 }
6027   \quark_if_no_value:nTF {#2}
6028   { \__xeCJK_copy_sub_family:n {#1} }
6029   { \xeCJK_set_family:onn \l__xeCJK_sub_family_name_tl #2 }
6030 }
6031 \cs_new_protected:Npn \__xeCJK_copy_sub_family:n #1
6032 {
6033   \__xeCJK_check_family:o \l__xeCJK_sub_family_name_tl
6034   \prop_get:NoNT \g__xeCJK_family_font_name_prop
6035   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
6036   {
6037     \prop_gput:Noo \g__xeCJK_family_font_name_prop
6038     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
6039   }
6040   \prop_get:NoNT \g__xeCJK_family_font_options_prop
6041   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
6042   {
6043     \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
6044     \prop_gput:Noo \g__xeCJK_family_font_options_prop
6045     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist

```

```

6046     }
6047     \cs_gset_protected:cpe
6048     { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
6049     {
6050         \xeCJK_family_if_exist:et { \l__xeCJK_family_name_tl }
6051         {
6052             \__xeCJK_gset_family_nfss_cs:no
6053             { \l__xeCJK_sub_family_name_tl }
6054             { \exp_not:N \l__xeCJK_fontspec_family_tl }
6055         }
6056     }
6057 }

\__xeCJK_copy_family:nn 6058 \cs_new_protected:Npn \__xeCJK_copy_family:nn #1#2
\__xeCJK_copy_family:ee 6059 {
6060     \xeCJK_family_if_exist:nT {#2}
6061     {
6062         \prop_gput:Nno \g__xeCJK_family_name_prop
6063         {#1} \l__xeCJK_fontspec_family_tl
6064         \tl_map_inline:nn
6065         {
6066             \g__xeCJK_family_font_name_prop
6067             \g__xeCJK_family_font_options_prop
6068         }
6069         {
6070             \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
6071             { \prop_gput:Nno ##1 {#1} \l__xeCJK_tmp_tl }
6072         }
6073         \cs_gset_eq:cc
6074         { \__xeCJK_family_nfss_csname:n {#1} }
6075         { \__xeCJK_family_nfss_csname:n {#2} }
6076     }
6077 }
6078 \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

```

6079 \cs_new:Npn \__xeCJK_font_csname:n #1
6080 { xeCJK/#1/\f@series/\f@shape/\f@size }
6081 \tl_new:N \l_xeCJK_current_font_tl
6082 \tl_set:N \l_xeCJK_current_font_tl
6083 { \__xeCJK_font_csname:n { \CJK@family } }
6084 \cs_new_protected:Npn \xeCJK_select_font:
6085 {
6086     \__xeCJK_select_font:cn
6087     { \l_xeCJK_current_font_tl }
6088     { \l_xeCJK_family_tl }
6089 }
6090 \cs_new_protected:Npn \__xeCJK_select_font:Nn #1#2
6091 {
6092     \xeCJK_clear_fallback_font:
6093     \cs_if_exist:NF #1 { \__xeCJK_font_initial:Nn #1 {#2} }
6094     #1
6095 }
6096 \cs_generate_variant:Nn \__xeCJK_select_font:Nn { c }
6097 \tl_new:N \l__xeCJK_current_coord_tl
6098 \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)。

```

6099 \cs_new_protected:Npn \__xeCJK_font_initial:Nn #1#2
6100 {
6101     \group_begin:

```

```

6102     \__xeCJK_font_initial_hook:
6103     \__xeCJK_family_use:n {#2}
6104     \xeCJK_font_gset_to_current:N #1
6105     \group_end:
6106   }
6107   \cs_new_protected:Npn \__xeCJK_font_initial_hook:
6108   { \tl_use:N \g__xeCJK_font_initial_hook_tl }
6109   \cs_new_protected:Npn \__xeCJK_gadd_font_initial_hook:n
6110   { \tl_gput_right:Nn \g__xeCJK_font_initial_hook_tl }
6111   \tl_new:N \g__xeCJK_font_initial_hook_tl

```

切换标点符号字体。

```

\xeCJK_select_punct_font:
\l_xeCJK_current_punct_font_tl
6112 \cs_new_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
6113 \cs_new_protected:Npn \__xeCJK_select_punct_font_aux:
6114 {
6115   \__xeCJK_select_font:cn
6116   { \l_xeCJK_current_punct_font_tl }
6117   { \l_xeCJK_punct_family_tl }
6118 }
6119 \tl_new:N \CJK@punctfamily
6120 \tl_new:N \l_xeCJK_punct_family_tl
6121 \tl_new:N \l_xeCJK_current_punct_font_tl
6122 \tl_set:Nn \CJK@punctfamily { \CJK@family }
6123 \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
6124 \tl_set:Nn \l_xeCJK_current_punct_font_tl
6125 { \__xeCJK_font_csname:n { \CJK@punctfamily } }
6126 \cs_new_eq:NN \__xeCJK_select_font: \prg_do_nothing:
6127 \cs_new_eq:NN \__xeCJK_select_punct_font: \prg_do_nothing:

```

两个 CJK 分区之间的字体切换。

```

\__xeCJK_switch_font:nn
6128 \cs_new_protected:Npn \__xeCJK_switch_font:nn #1#2
6129 {
6130   \str_if_eq:nnF {#1} {#2}
6131   {
6132     \__xeCJK_info:nee { CJK-block } {#1} {#2}
6133     \str_if_eq:nnTF {#2} { CJK }
6134     { \xeCJK_select_font: }
6135     { \xeCJK_select_font:n {#2} }
6136   }
6137 }
6138 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~`#1'~to~`#2'. }

```

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

```

\xeCJK_select_font:n
\xeCJK_block_family:nn
6139 \cs_new_protected:Npn \xeCJK_select_font:n #1
6140 {
6141   \__xeCJK_select_font:cnn
6142   { \__xeCJK_font_csname:n { \CJK@family/#1 } }
6143   { \l_xeCJK_family_tl }
6144   {#1}
6145 }
6146 \cs_new_protected:Npn \__xeCJK_select_font:Nnn #1#2#3
6147 {
6148   \xeCJK_clear_fallback_font:
6149   \cs_if_exist:NF #1
6150   { \__xeCJK_block_font_initial:Nnn #1 {#2} {#3} }
6151   #1
6152 }
6153 \cs_generate_variant:Nn \__xeCJK_select_font:Nnn { c }
6154 \cs_new_protected:Npn \__xeCJK_block_font_initial:Nnn #1#2#3
6155 {
6156   \xeCJK_block_family:nn {#2} {#3}
6157   \__xeCJK_font_initial:Nn #1 { #2/#3 }
6158 }

```

```

6159 \cs_new_protected:Npn \xeCJK_block_family:nn #1#2
6160 {
6161   \xeCJK_family_if_exist:ef { #1/#2 }
6162   {
6163     \__xeCJK_copy_family:ee { #1/#2 }
6164     {
6165       \cs_if_exist:cTF
6166       { \__xeCJK_family_csname:n { \CJKfamilydefault/#2 } }
6167       { \CJKfamilydefault/#2 } {#1}
6168     }
6169   }
6170 }

\__xeCJK_family_csname:n 6171 \cs_new:Npn \__xeCJK_family_csname:n #1
\__xeCJK_family_nfss_csname:n 6172 { xeCJK/family/#1 }
\__xeCJK_family_use:n 6173 \cs_new:Npn \__xeCJK_family_nfss_csname:n #1
\__xeCJK_gset_family_nfss_cs:nn 6174 { xeCJK/family/nfss/#1 }
6175 \cs_new_protected:Npn \__xeCJK_family_use:n #1
6176 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
6177 \cs_new_protected:Npn \__xeCJK_gset_family_nfss_cs:nn #1#2
6178 {
6179   \prop_gput:Nnn \g__xeCJK_family_name_prop {#1} {#2}
6180   \cs_gset_protected:cpe
6181   { \__xeCJK_family_nfss_csname:n {#1} }
6182   { \__xeCJK_nfss_family:nn { \c__xeCJK_encoding_tl } {#2} }
6183 }
6184 \cs_generate_variant:Nn \__xeCJK_gset_family_nfss_cs:nn { no }

```

__xeCJK_nfss_family:nn 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。

```

6185 \cs_if_exist:NTF \fontseriesforce
6186 {
6187   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
6188   {
6189     \fontencoding {#1}
6190     \str_if_eq:eeF { \f@series } { \bfdefault }
6191     {
6192       \str_case:e:nn { \f@family }
6193       {
6194         { \rmdefault } { \__xeCJK_nfss_series:n { rm } }
6195         { \sfdefault } { \__xeCJK_nfss_series:n { sf } }
6196         { \ttdefault } { \__xeCJK_nfss_series:n { tt } }
6197       }
6198     }
6199     \fontfamily {#2}
6200     \selectfont
6201   }
6202   \cs_new_protected:Npn \__xeCJK_nfss_series:n #1
6203   {
6204     \str_if_eq:eeT { \f@series } { \use:c { bfseries@#1 } }
6205     { \fontseriesforce { \bfdefault } }
6206   }
6207 }
6208 {
6209   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
6210   {
6211     \fontencoding {#1}
6212     \tl_set:Nn \f@family {#2}
6213     \selectfont
6214   }
6215 }

```

```

\xeCJK_family_if_exist:nTF 6216 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:n #1 { T , F , TF }
6217 {
6218   \prop_get:NnNTF \g__xeCJK_family_name_prop
6219   {#1} \l__xeCJK_fontspec_family_tl
6220   { \prg_return_true: }
6221   {

```

```

6222     \exp_args:Ne \cs_if_exist_use:cTF
6223     { \_xeCJK_family_csname:n {#1} }
6224     { \prg_return_true: }
6225     { \prg_return_false: }
6226   }
6227 }
6228 \prg_generate_conditional_variant:Nnn \xeCJK_family_if_exist:n { e } { T , F , TF }

```

\CJKfamily 用于切换 CJK 字体族。

```

6229 \NewDocumentCommand \CJKfamily { t+ t- m }
6230 {
6231   \xeCJK_family:NNe #1 #2 {#3}
6232   \tex_ignorespaces:D
6233 }
6234 \cs_new_protected:Npn \xeCJK_family:NNn #1#2#3
6235 {
6236   \tl_if_blank:nTF {#3}
6237   {
6238     \bool_if:NF #1 { \bool_if:NF #2 { \use_none:nn } }
6239     \xeCJK_family_if_exist_use:e { \l_xeCJK_family_tl }
6240   }
6241   {
6242     \bool_if:NTF #2
6243     { \xeCJK_family_if_exist_use:n {#3} }
6244     {
6245       \xeCJK_family_if_exist:nTF {#3}
6246       {
6247         \tl_set:Nn \l_xeCJK_family_tl {#3}
6248         \tl_set_eq:NN \CJK@family \l_xeCJK_fontspec_family_tl
6249         \bool_if:NT #1 { \_xeCJK_family_use:n {#3} }
6250       }
6251       { \_xeCJK_family_unknown_warning:n {#3} }
6252     }
6253   }
6254 }
6255 \cs_generate_variant:Nn \xeCJK_family:NNn { NNe }
6256 \cs_new_protected:Npn \xeCJK_switch_family:n #1
6257 {
6258   \xeCJK_family_if_exist:nTF {#1}
6259   {
6260     \tl_set:Nn \l_xeCJK_family_tl {#1}
6261     \tl_set_eq:NN \CJK@family \l_xeCJK_fontspec_family_tl
6262   }
6263   { \_xeCJK_family_unknown_warning:n {#1} }
6264 }
6265 \cs_generate_variant:Nn \xeCJK_switch_family:n { e , o }

```

PunctFamily 设置汉字标点符号的字体。

```

6266 \keys_define:nn { xeCJK / options }
6267 {
6268   PunctFamily .choice: ,
6269   PunctFamily .value_required:n = { true } ,
6270   PunctFamily / false .code:n =
6271   {
6272     \tl_set:Nn \CJK@punctfamily { \CJK@family }
6273     \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
6274     \xeCJK_cs_clear:N \_xeCJK_select_font:
6275     \xeCJK_cs_clear:N \_xeCJK_select_punct_font:
6276     \cs_set_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
6277   } ,
6278   PunctFamily / unknown .code:n =
6279   { \xeCJK_punct_family:e {#1} } ,
6280 }
6281 \cs_new_protected:Npn \xeCJK_punct_family:n #1
6282 {

```

```

6283 \xeCJK_family_if_exist:nTF {#1}
6284 {
6285     \tl_set:Nn \l_xeCJK_punct_family_tl {#1}
6286     \tl_set_eq:NN \CJK@punctfamily \l_xeCJK_fontspec_family_tl
6287     \cs_set_eq:NN \_xeCJK_select_font: \xeCJK_select_font:
6288     \cs_set_eq:NN \_xeCJK_select_punct_font: \_xeCJK_select_punct_font_aux:
6289     \cs_set_eq:NN \xeCJK_select_punct_font: \_xeCJK_select_punct_font:
6290 }
6291 { \_xeCJK_family_unknown_warning:n {#1} }
6292 }
6293 \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。

```
6294 \tl_new:N \l_xeCJK_family_tl
```

\CJK@family 用于保存实际的字体族名称。

```
6295 \tl_new:N \CJK@family
```

```

\_xeCJK_gobble_CJKfamily: 6296 \cs_new_protected:Npn \_xeCJK_gobble_CJKfamily:
6297 { \cs_set_eq:NN \CJKfamily \_xeCJK_gobble_CJKfamily:wn }
6298 \NewExpandableDocumentCommand \_xeCJK_gobble_CJKfamily:wn { t+ t- m } { }

```

```

\xeCJK_family_if_exist_use:n 6299 \cs_new_protected:Npn \xeCJK_family_if_exist_use:n #1
6300 {
6301     \xeCJK_family_if_exist:nTF {#1}
6302     { \_xeCJK_family_use:n {#1} }
6303     { \_xeCJK_family_unknown_warning:n {#1} }
6304 }
6305 \cs_generate_variant:Nn \xeCJK_family_if_exist_use:n { e }

```

```

\_xeCJK_family_unknown_warning:n 6306 \cs_new_protected:Npn \_xeCJK_family_unknown_warning:n #1
6307 {
6308     \prop_if_empty:NF \g__xeCJK_family_font_name_prop
6309     {
6310         \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
6311         {
6312             \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
6313             \_xeCJK_warning:ne { CJKfamily-Unknown } {#1}
6314         }
6315     }
6316 }
6317 \seq_new:N \g__xeCJK_unknown_family_seq
6318 \_xeCJK_msg_new:nn { CJKfamily-Unknown }
6319 {
6320     Unknown~CJK~family~~\_xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
6321     Try~to~use~~\_xeCJK_msg_def_family_map:n {#1}'~to~define~it.
6322 }
6323 \cs_new:Npn \_xeCJK_msg_def_family_map:n #1
6324 {
6325     \str_case_e:nnF {#1}
6326     {
6327         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
6328         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
6329         \CJKttdefault { \token_to_str:N \setCJKmonofont }
6330     }
6331     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
6332     [<...>]\{<...>\}
6333 }
6334 \cs_new:Npn \_xeCJK_msg_family_map:n #1
6335 {
6336     \str_case_e:nnF {#1}
6337     {
6338         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
6339         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
6340         \CJKttdefault { \token_to_str:N \CJKttdefault }
6341     }
6342     {#1}
6343 }

```

`\_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。

```

6344 \cs_new_protected:Npn \_xeCJK_pass_args:nnnn #1#2#3#4
6345 {
6346   \tl_if_novalue:nTF {#2}
6347   { \_xeCJK_post_arg:w {#1} {#3} {#4} }
6348   {
6349     \use:e { #1 {#2} {#3} }
6350     #4
6351   }
6352 }
6353 \NewDocumentCommand \_xeCJK_post_arg:w { m m m O { } }
6354 {
6355   \use:e { #1 {#4} {#2} }
6356   #3
6357 }

```

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

```

\setCJKsansfont
\setCJKmonofont
6358 \NewDocumentCommand \setCJKmainfont { o m }
6359 {
6360   \_xeCJK_pass_args:nnnn
6361   { \xeCJK_set_family:nnn { \CJKrmdefault } } {#1} {#2}
6362   { \_xeCJK_preamble_family:n { \CJKrmdefault } }
6363 }
6364 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
6365 \NewDocumentCommand \setCJKsansfont { o m }
6366 {
6367   \_xeCJK_pass_args:nnnn
6368   { \xeCJK_set_family:nnn { \CJKsfdefault } } {#1} {#2}
6369   { \_xeCJK_preamble_family:n { \CJKsfdefault } }
6370 }
6371 \NewDocumentCommand \setCJKmonofont { o m }
6372 {
6373   \_xeCJK_pass_args:nnnn
6374   { \xeCJK_set_family:nnn { \CJKttdefault } } {#1} {#2}
6375   { \_xeCJK_preamble_family:n { \CJKttdefault } }
6376 }

6377 \@onlypreamble \setCJKmainfont
6378 \@onlypreamble \setCJKmathfont
6379 \@onlypreamble \setCJKsansfont
6380 \@onlypreamble \setCJKmonofont
6381 \@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。

```

6382 \cs_new_protected:Npn \_xeCJK_preamble_family:n #1
6383 { \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
6384 \NewDocumentCommand \setCJKfamilyfont { m o m }
6385 {
6386   \_xeCJK_pass_args:nnnn
6387   { \xeCJK_set_family:nnn {#1} } {#2} {#3}
6388   { }
6389 }
6390 \_xeCJK_msg_new:nn { command-already-defined }
6391 { Control~sequence~#1~is~already~defined. }
6392 \NewDocumentCommand \newCJKfontfamily { o m o m }
6393 {
6394   \tl_set:Nx \l__xeCJK_tmp_tl
6395   { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
6396   \cs_if_exist:NTF #2
6397   { \_xeCJK_error:ne { command-already-defined } { \token_to_str:N #2 } }
6398   {

```

```

6399         \cs_set_protected:Npe #2
6400         { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
6401     }
6402     \__xeCJK_pass_args:nnnn
6403     { \xeCJK_set_family:nnn { \l__xeCJK_tmp_tl } } {#3} {#4}
6404     { }
6405 }
6406 \NewDocumentCommand \CJKfontspec { o m }
6407 {
6408     \__xeCJK_pass_args:nnnn
6409     { \xeCJK_fontspec:nn } {#1} {#2}
6410     { \tex_ignorespaces:D }
6411 }

\xeCJK_fontspec:nn 6412 \cs_new_protected:Npn \xeCJK_fontspec:nn #1#2
6413 {
6414     \prop_get:NnNTF \g__xeCJK_fontspec_prop
6415     { CJKfontspec/#1/#2/id } \l_xeCJK_family_tl
6416     { \xeCJK_switch_family:o { \l_xeCJK_family_tl } }
6417     {
6418         \__xeCJK_fontspec:enn
6419         { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + 1 } ) }
6420         {#1} {#2}
6421     }
6422 }
6423 \cs_new_protected:Npn \__xeCJK_fontspec:nnn #1#2#3
6424 {
6425     \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
6426     \xeCJK_set_family:nnn {#1} {#2} {#3}
6427     \xeCJK_switch_family:n {#1}
6428 }
6429 \cs_generate_variant:Nn \xeCJK_fontspec:nn { oo }
6430 \cs_generate_variant:Nn \__xeCJK_fontspec:nnn { e }
6431 \prop_new:N \g__xeCJK_fontspec_prop

\defaultCJKfontfeatures 分别用于设置 CJK 字体的默认属性和增加当前 CJK 字体的属性。
\addCJKfontfeatures
6432 \clist_new:N \g__xeCJK_default_features_clist
6433 \NewDocumentCommand \defaultCJKfontfeatures { m }
6434 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
6435 \@onlypreamble \defaultCJKfontfeatures
6436 \NewDocumentCommand \addCJKfontfeatures { s O { } m }
6437 {
6438     \xeCJK_add_font_features:Nee #1 {#2} {#3}
6439     \tex_ignorespaces:D
6440 }
6441 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\xeCJK_add_font_features:Nnn 6442 \cs_new_protected:Npn \xeCJK_add_font_features:Nnn #1#2#3
6443 {
6444     \prop_get:NoNTF \g__xeCJK_family_font_name_prop
6445     \l_xeCJK_family_tl \l__xeCJK_font_name_tl
6446     {
6447         \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
6448         \seq_map_inline:Nn \g__xeCJK_sub_key_seq
6449         { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
6450         \seq_clear:N \l__xeCJK_sub_key_seq
6451         \clist_clear:N \l__xeCJK_add_block_features_clist
6452         \clist_map_function:nN {#2} \__xeCJK_add_sub_features:n
6453         \bool_lazy_and:nnT
6454         {#1}
6455         { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
6456         {
6457             \seq_map_function:NN
6458             \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
6459         }
6460     }
6461     \prop_get:NoNT \g__xeCJK_family_font_options_prop
6462     \l_xeCJK_family_tl \l__xeCJK_font_options_clist

```



```

6462     {
6463         \bool_lazy_or:nnT
6464         { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
6465         {#1}
6466         {
6467             \clist_concat:NNN \l__xeCJK_font_options_clist
6468             \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
6469         }
6470         \clist_concat:NNN \l__xeCJK_font_options_clist
6471         \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
6472     }
6473     \xeCJK_fontspec:oo \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
6474 }
6475 { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
6476 }
6477 \cs_new_protected:Npn \__xeCJK_add_sub_features:n #1
6478 {
6479     \seq_if_in:NnTF \g__xeCJK_sub_key_seq {#1}
6480     {
6481         \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
6482         \__xeCJK_add_sub_class_features:n {#1}
6483     }
6484     { \__xeCJK_warning:ne { SubBlock-undefined } {#1} }
6485 }
6486 \clist_new:N \l__xeCJK_add_font_features_clist
6487 \clist_new:N \l__xeCJK_add_block_features_clist
6488 \cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nee , Nne }
6489 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
6490 {
6491     \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
6492     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
6493 }
\__xeCJK_add_sub_class_features:n 6494 \cs_new_protected:Npn \__xeCJK_add_sub_class_features:n #1
6495 {
6496     \prop_get:NoNTF \g__xeCJK_family_font_name_prop
6497     { \l__xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
6498     {
6499         \prop_get:NoN \g__xeCJK_family_font_options_prop
6500         { \l__xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
6501     }
6502     {
6503         \prop_get:NeNTF \g__xeCJK_family_font_name_prop
6504         { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
6505         {
6506             \prop_get:NeN \g__xeCJK_family_font_options_prop
6507             { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
6508         }
6509         {
6510             \prop_get:NoN \g__xeCJK_family_font_options_prop
6511             \l__xeCJK_family_tl \l__xeCJK_sub_font_options_clist
6512             \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl
6513         }
6514     }
6515     \clist_concat:NNN \l__xeCJK_sub_font_options_clist
6516     \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
6517     \clist_put_right:Ne \l__xeCJK_add_block_features_clist
6518     {
6519         #1 =
6520         {
6521             [ \exp_not:o \l__xeCJK_sub_font_options_clist ]
6522             { \exp_not:o \l__xeCJK_sub_font_name_tl }
6523         }
6524     }
6525 }
6526 \cs_generate_variant:Nn \prop_get:NnN { Ne }
6527 \prg_generate_conditional_variant:Nnn \prop_get:NnN { Ne } { T , TF }

```

```

LoadFandol 6528 \keys_define:nn { xeCJK / options }
6529 { LoadFandol .bool_gset:N = \g__xeCJK_fandol_bool }
6530 \cs_new_protected:Npn \__xeCJK_load_fandol:
6531 {
6532   \xeCJK_set_family:enn { \CJKrmdefault }
6533   { Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular }
6534   { FandolSong-Regular }
6535   \xeCJK_set_family:enn { \CJKsfdefault }
6536   { Extension = .otf , BoldFont = FandolHei-Bold } { FandolHei-Regular }
6537   \xeCJK_set_family:enn { \CJKttdefault }
6538   { Extension = .otf } { FandolFang-Regular }
6539 }

```

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。

```

6540 \__xeCJK_at_end_preamble:n
6541 {
6542   \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
6543   {
6544     \group_begin:
6545     \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
6546     \tl_gset:Ne \CJKfamilydefault
6547     {
6548       \str_case:onF { \familydefault }
6549       {
6550         { \rmdefault } { \exp_not:N \CJKrmdefault }
6551         { \sfdefault } { \exp_not:N \CJKsfdefault }
6552         { \ttdefault } { \exp_not:N \CJKttdefault }
6553       }
6554       { \CJKfamilydefault }
6555     }
6556     \group_end:
6557   }
6558   \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
6559   {
6560     \bool_if:NTF \g__xeCJK_fandol_bool
6561     {
6562       \__xeCJK_warning:n { fandol }
6563       \__xeCJK_load_fandol:
6564       \xeCJK_ensure_default_family:
6565     }
6566     { \__xeCJK_warning:ne { no-CJKfamily } { \CJKfamilydefault } }
6567   }
6568   { \xeCJK_ensure_default_family: }
6569 }
6570 \cs_new_protected:Npn \xeCJK_ensure_default_family:
6571 {
6572   \xeCJK_family_if_exist:eF { \CJKfamilydefault }
6573   {
6574     \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
6575     \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
6576     { \use:n }
6577     {
6578       \xeCJK_family_if_exist:eTF { \CJKrmdefault }
6579       { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
6580     }
6581   }
6582   \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
6583   {
6584     \prop_map_break:n
6585     { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
6586   }
6587 }

```

```

6588     \__xeCJK_warning:nne { CJKfamilydefault-undefined }
6589     { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
6590   }
6591   \xeCJK_switch_family:e { \CJKfamilydefault }
6592   \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
6593 }
6594 \__xeCJK_msg_new:nn { no-CJKfamily }
6595 {
6596   It~seems~that~you~have~not~declared~a~CJKfamily.\\
6597   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
6598   \__xeCJK_msg_def_family_map:n {#1}'\\
6599   in~the~preamble~to~declare~the~default~CJKfamily.\\
6600 }
6601 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
6602 {
6603   Undefined~CJK~default~family~\__xeCJK_msg_family_map:n {#1}'~
6604   has~been~replaced~by~\__xeCJK_msg_family_map:n {#2}'\\.\\
6605   Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
6606 }
6607 \__xeCJK_msg_new:nn { fandol }
6608 {
6609   Fandol~is~being~set~as~the~default~font~for~CJK~text.\\
6610   Please~make~sure~it~has~been~properly~installed.
6611 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```
6612 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }
```

\setCJKmathfont 设置 CJK 数学字体。

```

6613 \NewDocumentCommand \setCJKmathfont { o m }
6614 {
6615   \__xeCJK_pass_args:nnnn
6616   { \xeCJK_set_family:nnn { \c__xeCJK_math_tl } } {#1} {#2}
6617   { }
6618 }
6619 \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。

```

6620 \cs_new_protected:Npn \xeCJK_set_mathfont:
6621 {
6622   \cs_if_exist_use:N \__xeCJK_save_um_char:
6623   \xeCJK_family_if_exist:eTF { \c__xeCJK_math_tl }
6624   { \__xeCJK_set_mathfont_aux: }
6625   {
6626     \xeCJK_family_if_exist:eT { \CJKfamilydefault }
6627     {
6628       \__xeCJK_copy_family:ee { \c__xeCJK_math_tl } { \CJKfamilydefault }
6629       \__xeCJK_set_mathfont_aux:
6630     }
6631   }
6632   \cs_if_exist_use:N \__xeCJK_restore_um_char:
6633 }
6634 \cs_new_protected:Npn \__xeCJK_set_mathfont_aux:
6635 {
6636   \tl_const:Nn \c__xeCJK_math_family_tl
6637   { \l__xeCJK_fontspec_family_tl }
6638   \xeCJK_declare_mathfont:ee
6639   { \c__xeCJK_math_tl }
6640   { \c__xeCJK_math_family_tl }
6641   \int_const:Nn \c__xeCJK_math_fam_int
6642   { \use:c { sym \c__xeCJK_math_tl } }
6643   \clist_gconcat:NNN \g__xeCJK_math_chars_clist

```

```

6644 \g__xeCJK_CJK_range_clist \g__xeCJK_FullLeft_range_clist
6645 \clist_gconcat:NNN \g__xeCJK_math_chars_clist
6646 \g__xeCJK_math_chars_clist \g__xeCJK_FullRight_range_clist
6647 \xeCJK_gset_mathcode:Nn \g__xeCJK_math_chars_clist
6648 { \c__xeCJK_math_fam_int }
6649 \xeCJK_set_mathfont_block:
6650 }
6651 \clist_new:N \g__xeCJK_math_chars_clist
6652 \prop_new:N \g__xeCJK_fam_prop

```

\xeCJK_set_mathfont_block: 分区数学字体。

```

6653 \cs_new_protected:Npn \xeCJK_set_mathfont_block:
6654 {
6655   \seq_if_empty:NF \g__xeCJK_CJK_sub_class_seq
6656   {
6657     \seq_map_function:NN
6658     \g__xeCJK_CJK_sub_class_seq
6659     \xeCJK_set_mathfont_block:n
6660   }
6661 }
6662 \cs_new_protected:Npn \xeCJK_set_mathfont_block:n #1
6663 {
6664   \xeCJK_block_family:nn { \c__xeCJK_math_tl } {#1}
6665   \prop_get:NoNTF \g__xeCJK_fam_prop
6666   \l__xeCJK_fontspec_family_tl \l__xeCJK_tmp_tl
6667   { \int_set:Nn \l__xeCJK_fam_int { \l__xeCJK_tmp_tl } }
6668   {
6669     \xeCJK_declare_mathfont:ee
6670     { \c__xeCJK_math_tl / #1 }
6671     { \l__xeCJK_fontspec_family_tl }
6672     \__xeCJK_set_mathfont_block_aux:cn
6673     { sym \c__xeCJK_math_tl / #1 } {#1}
6674   }
6675   \xeCJK_gset_mathcode:cn { g__xeCJK_CJK/#1_range_clist } { \l__xeCJK_fam_int }
6676 }
6677 \cs_new_protected:Npn \__xeCJK_set_mathfont_block_aux:Nn #1#2
6678 {
6679   \int_set_eq:NN \l__xeCJK_fam_int #1
6680   \prop_gput:Nnn \g__xeCJK_block_fam_prop {#2} {#1}
6681 }
6682 \int_new:N \l__xeCJK_fam_int
6683 \prop_new:N \g__xeCJK_block_fam_prop
6684 \cs_generate_variant:Nn \__xeCJK_set_mathfont_block_aux:Nn { c }

```

\xeCJK_declare_mathfont:nn 注意从 L^AT_EX₂_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。

```

6685 \cs_new_protected:Npn \xeCJK_declare_mathfont:nn #1#2
6686 {
6687   \xeCJK_declare_symbol_font:nnnnn {#1} { \c__xeCJK_encoding_tl }
6688   {#2} { \mddefault } { \shapedefault }
6689   \cs_if_free:cF
6690   { \c__xeCJK_encoding_tl/#2/\bfdefault/\shapedefault }
6691   {
6692     \SetSymbolFont {#1} { bold } { \c__xeCJK_encoding_tl }
6693     {#2} { \bfdefault } { \shapedefault }
6694   }
6695   \prop_gput:Nne \g__xeCJK_fam_prop {#2} { \exp_not:c { sym #1 } }
6696 }
6697 \cs_generate_variant:Nn \prop_put:Nnn { Nne }
6698 \cs_generate_variant:Nn \prop_gput:Nnn { Nne }
6699 \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。

```

6700 \cs_new_protected:Npn \xeCJK_declare_symbol_font:nnnnn #1

```

```

6701 { \__xeCJK_declare_symbol_font:cnnnn { sym #1 } }
6702 \cs_new_protected:Npn \__xeCJK_declare_symbol_font:Nnnnn #1
6703 {
6704   \xeCJK_new_fam:N #1
6705   \xeCJK_new_symbol_font:Nnnnn #1
6706 }
6707 \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。

```

6708 \cs_new_protected:Npn \xeCJK_new_fam:N #1
6709 {
6710   \int_compare:nNtF
6711     { \g__xeCJK_fam_allocation_int } > { \g__xeCJK_fam_bottom_int }
6712     {
6713       \int_set_eq:NN \allocationnumber \g__xeCJK_fam_allocation_int
6714       \int_const:Nn #1 { \allocationnumber }
6715       \iow_log:e
6716       {
6717         \token_to_str:N #1 =
6718         \token_to_str:N \mathgroup \int_use:N \allocationnumber
6719       }
6720       \int_gdecr:N \g__xeCJK_fam_allocation_int
6721     }
6722     { \__xeCJK_error:n { fam-exhausted } }
6723 }
6724 \tex_countdef:D \g__xeCJK_fam_bottom_int = 18 ~
6725 \int_new:N \g__xeCJK_fam_allocation_int
6726 \int_gset:Nn \g__xeCJK_fam_allocation_int { 255 }
6727 \__xeCJK_msg_new:nn { fam-exhausted }
6728 { 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
6729 \cs_new_protected:Npn \xeCJK_new_symbol_font:Nnnnn #1#2#3#4#5
6730 { \__xeCJK_new_symbol_font:Nc #1 { #2/#3/#4/#5 } }
6731 \cs_new_protected:Npn \__xeCJK_new_symbol_font:NN #1#2
6732 {
6733   \tl_put_right:Nn \group@list { \group@elt #1 #2 }
6734   \cs_set:Npn \version@elt ##1
6735     { \tl_put_right:Nn ##1 { \getanddefine@fonts #1 #2 } }
6736   \version@list
6737 }
6738 \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
6739 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nn #1#2
6740 {
6741   \clist_map_inline:Nn #1
6742     {
6743       \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn { ##1 }
6744       { 0 } {#2}
6745     }
6746 }
6747 \cs_generate_variant:Nn \xeCJK_gset_mathcode:Nn { c }
6748 \cs_new_protected:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
6749 {
6750   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
6751   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
6752   {
6753     \xeCJK_gset_mathcode:Nnn \l__xeCJK_begin_int {#3} {#4}
6754     \int_incr:N \l__xeCJK_begin_int
6755   }
6756 }
6757 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nnn #1#2#3
6758 { \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。

```

6759 \int_new:N \l__xeCJK_verb_case_int
6760 \keys_define:nn { xeCJK / options }
6761 {
6762   Verb .choices:nn =
6763     { true , env+ , env , false }
6764     { \int_set_eq:NN \l__xeCJK_verb_case_int \l_keys_choice_int } ,
6765   Verb .default:n = { env }
6766 }
6767 \cs_new_protected:Npn \__xeCJK_verb_font_hook:
6768 {
6769   \if_case:w \l__xeCJK_verb_case_int
6770   \or:
6771     \__xeCJK_nobreak_skip_zero:
6772   \or:
6773     \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
6774       { \xeCJKVerbAddon }
6775       { \__xeCJK_nobreak_skip: }
6776   \or:
6777     \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
6778       { \xeCJKVerbAddon }
6779       { \__xeCJK_nobreak_skip_zero: }
6780   \fi:
6781 }
6782 \__xeCJK_after_preamble:n
6783 {
6784   \cs_set_protected:Npe \verbatim@font
6785     { \exp_not:o { \verbatim@font } \__xeCJK_verb_font_hook: }
6786 }
\__xeCJK_nobreak_skip_zero: 6787 \cs_new_protected:Npn \__xeCJK_nobreak_skip_zero:
\__xeCJK_nobreak_skip: 6788 {
6789   \__xeCJK_reset_shipout_skip:
6790   \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
6791   \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
6792   \tl_put_right:Nn \l__xeCJK_reset_shipout_skip_hook_tl
6793     {
6794       \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
6795       \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
6796     }
6797   \xeCJK_cs_clear:N \CJKglue
6798   \xeCJK_cs_clear:N \CJKecglue
6799   \xeCJK_cs_clear:N \xeCJK_check_for_glue:
6800   \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \xeCJK_class_group_end:
6801   \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
6802   \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
6803 }
6804 \cs_new_protected:Npn \__xeCJK_nobreak_skip:
6805 {
6806   \__xeCJK_reset_shipout_skip:
6807   \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_ccglue_skip
6808   \skip_if_eq:nnTF { \l__xeCJK_ccglue_skip } { \c_zero_skip }
6809     { \xeCJK_cs_clear:N \CJKglue }
6810     { \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue: }
6811   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
6812   \skip_if_eq:nnTF { \l__xeCJK_ecglue_skip } { \c_zero_skip }
6813     { \xeCJK_cs_clear:N \CJKecglue }
6814     { \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue: }
6815   \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
6816   \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
6817 }
6818 \cs_new_protected:Npn \__xeCJK_nobreak_ccglue:
6819 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ccglue_skip }

```

```

6820 \cs_new_protected:Npn \__xeCJK_nobreak_ecglue:
6821 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ecglue_skip }

\__xeCJK_reset_shipout_skip: 6822 \cs_new_protected:Npn \__xeCJK_reset_shipout_skip:
6823 {
6824   \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
6825   \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
6826   \cs_set_eq:NN \__xeCJK_shipout_punct_hskip:n \__xeCJK_punct_hskip:n
6827   \cs_set_eq:NN
6828     \__xeCJK_shipout_punct_breakable_kern:n \__xeCJK_punct_breakable_kern:n
6829   \tl_set:Nx \l__xeCJK_off_verb_addon_tl
6830   {
6831     \bool_if:NTF \l__xeCJK_xecglue_bool
6832       { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
6833       { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
6834     \exp_not:n
6835     {
6836       \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
6837       \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
6838       \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_shipout_punct_hskip:n
6839       \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n
6840         \__xeCJK_shipout_punct_breakable_kern:n
6841       \l__xeCJK_reset_shipout_skip_hook_tl
6842     }
6843   }
6844   \xeCJK_add_to_shipout:n { \l__xeCJK_off_verb_addon_tl }
6845   \keys_set:nn { xeCJK / options } { xCJKecglue = false }
6846 }
6847 \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。

```

6848 \NewDocumentCommand \xeCJKVerbAddOn { }
6849 {
6850   \int_compare:nNnF \tex_currentgrouplevel:D = \c_zero_int
6851   {
6852     \bool_if:NF \l__xeCJK_listings_env_bool
6853     {
6854       \dim_compare:nNnTF
6855         { \tex_fontdimen:D 2 ~ \tex_font:D } =
6856         { \tex_fontcharwd:D \tex_font:D \c__xeCJK_mono_letter_int }
6857         {
6858           \__xeCJK_set_verb_exspace:
6859           \__xeCJK_verb_addon:
6860         }
6861         {
6862           \int_if_odd:nTF { \l__xeCJK_verb_case_int }
6863             { \__xeCJK_nobreak_skip_zero: }
6864             { \__xeCJK_nobreak_skip: }
6865         }
6866     }
6867   }
6868 }
6869 \int_const:Nn \c__xeCJK_mono_letter_int { 77 }
6870 \bool_new:N \l__xeCJK_listings_env_bool
6871 \NewDocumentCommand \xeCJKOffVerbAddOn { }
6872 { \tl_use:N \l__xeCJK_off_verb_addon_tl }
6873 \tl_new:N \l__xeCJK_off_verb_addon_tl
6874 \cs_new_protected:Npn \__xeCJK_verb_addon:
6875 {
6876   \bool_if:NF \l__xeCJK_verb_addon_bool
6877   { \__xeCJK_verb_addon_action: }
6878   \skip_if_eq:nnTF { \l__xeCJK_verb_exspace_skip } { \c_zero_skip }
6879   {

```

```

6880 \xeCJK_cs_clear:N \CJKglue
6881 \xeCJK_cs_clear:N \CJKecglue
6882 }
6883 {
6884 \skip_set_eq:NN \l__xeCJK_ccglue_skip \l__xeCJK_verb_exspace_skip
6885 \skip_set:Nn \l__xeCJK_ecglue_skip { \l__xeCJK_verb_exspace_skip / 2 }
6886 \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue:
6887 \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue:
6888 }
6889 \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
6890 \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_CJK_and_Boundary:w
6891 }
6892 \cs_new_protected:Npn \__xeCJK_verb_addon_action:
6893 {
6894 \bool_set_true:N \l__xeCJK_verb_addon_bool
6895 \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
6896 \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
6897 \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
6898 \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
6899 \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
6900 \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
6901 \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
6902 \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
6903 \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
6904 \cs_set_protected:Npe \xeCJKOffVerbAddon
6905 {
6906 \__xeCJK_reset_char_class:n { FullLeft }
6907 \__xeCJK_reset_char_class:n { FullRight }
6908 \__xeCJK_reset_char_class:n { HalfLeft }
6909 \__xeCJK_reset_char_class:n { HalfLeft }
6910 \__xeCJK_reset_char_class:n { NormalSpace }
6911 \bool_if:NTF \l__xeCJK_xecglue_bool
6912 { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
6913 { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
6914 \exp_not:n
6915 {
6916 \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
6917 \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
6918 \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
6919 \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
6920 }
6921 }
6922 \xeCJK_add_to_shipout:n { \xeCJKOffVerbAddon }
6923 \keys_set:nn { xeCJK / options } { xCJKecglue = false }
6924 }
6925 \cs_new_protected:Npn \__xeCJK_verb_CJK_and_Boundary:w
6926 { \xeCJK_class_group_end: \CJKecglue }
6927 \cs_new_protected:Npn \__xeCJK_reset_char_class:n #1
6928 {
6929 \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
6930 \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
6931 { \tex_XeTeXcharclass:D ##1 = \l__xeCJK_tmp_int }
6932 }
6933 \bool_new:N \l__xeCJK_verb_addon_bool
6934 \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。

```

6935 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:
6936 {
6937 \tl_if_exist:cTF { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
6938 {
6939 \skip_set:Nn \l__xeCJK_verb_exspace_skip
6940 { \use:c { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size } }
6941 }
6942 {

```



```

6943 \tl_set:Nn \l__xeCJK_current_coor_tl { \CJK@family/\curr@fontshape }
6944 \prop_get:NnTF \g__xeCJK_scale_family_prop
6945 \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
6946 {
6947   \xeCJK_switch_family:o { \l_xeCJK_family_tl }
6948   \skip_zero:N \l__xeCJK_verb_exspace_skip
6949 }
6950 {
6951   \group_begin: \xeCJK_select_font: \exp_args:NNo \group_end:
6952   \__xeCJK_set_verb_exspace:n
6953   { \dim_use:N \tex_fontcharwd:D \tex_font:D "4E00 ~ }
6954 }
6955 }
6956 }
6957 \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。

```

6958 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:n #1
6959 {
6960   \skip_set:Nn \l__xeCJK_verb_exspace_skip
6961   { 2 \tex_fontdimen:D 2 ~ \tex_font:D - #1 }
6962   \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
6963   {
6964     \skip_zero:N \l__xeCJK_verb_exspace_skip
6965     \exp_args:Nee \__xeCJK_set_verb_scale:nn
6966     { \dim_to_fp:n { 2 \tex_fontdimen:D 2 ~ \tex_font:D } }
6967     { \dim_to_fp:n {#1} }
6968   }
6969   {
6970     \tl_const:ce { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
6971     { \skip_use:N \l__xeCJK_verb_exspace_skip }
6972   }
6973 }

```

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

```

6974 \cs_new_protected:Npn \__xeCJK_set_verb_scale:nn #1#2
6975 {
6976   \fp_set:Nn \l__xeCJK_scale_factor_fp { #1 / #2 }
6977   \__xeCJK_warning:nee { scale-factor }
6978   { \fp_eval:n { trunc ( \l__xeCJK_scale_factor_fp , 4 ) } }
6979   { \fp_eval:n { ceil ( #2 / #1 , 4 ) } }
6980   \xeCJK_add_font_features:Nne \c_true_bool
6981   { } { Scale = { \fp_use:N \l__xeCJK_scale_factor_fp } }
6982   \prop_gput:Noo \g__xeCJK_scale_family_prop
6983   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
6984 }
6985 \__xeCJK_msg_new:nn { scale-factor }
6986 {
6987   \token_to_str:N \xeCJKVerbAddon'~may~not~work~properly.\\
6988   You~may~set~`Scale=#1'~to~CJKfamily~
6989   \__xeCJK_msg_family_map:n { \l_xeCJK_family_tl }',\\
6990   or~set~`Scale=#2'~to~family~
6991   \str_if_eq:eeTF \f@family \ttdefault
6992   { \token_to_str:N \ttdefault } { \f@family }'.
6993 }
6994 \fp_new:N \l__xeCJK_scale_factor_fp
6995 \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 T_EX 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。

```

6996 \cs_new_protected:Npn \xeCJK_setup_visible_space:
6997 {
6998   \xeCJK_make_boundary:
6999   \xeCJK_glyph_if_exist:Ntf { ~~~~2423 }
7000   { \tl_set:Nn \l__xeCJK_visible_space_tl { ~~~~2423 } }

```

```

7001     {
7002         \int_compare:nNnTF { \tex_XeTeXfonttype:D \tex_font:D } = \c_zero_int
7003         {
7004             \tl_set:Nc \l__xeCJK_visible_space_tl
7005             {
7006                 \str_if_eq:eeTF { \f@family } { \ttdefault }
7007                 { \c_catcode_other_space_tl }
7008                 { \exp_not:N \textvisiblespace }
7009             }
7010         }
7011         { \__xeCJK_visible_space_fallback: }
7012     }
7013     \cs_set_eq:NN \xobeysp \l__xeCJK_visible_space_tl
7014 }
7015 \tl_new:N \l__xeCJK_visible_space_tl
7016 \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。

```

7017 \cs_new_protected:Npn \__xeCJK_visible_space_fallback:
7018 {
7019     \exp_args:Nc \__xeCJK_visible_space_fallback_auxi:N
7020     { xeCJK/space/\curr@fontshape/\f@size }
7021 }
7022 \cs_new_protected:Npn \__xeCJK_visible_space_fallback_auxi:N #1
7023 {
7024     \cs_if_exist:NF #1
7025     { \__xeCJK_visible_space_fallback_auxii:N #1 }
7026     \tl_set:Nn \l__xeCJK_visible_space_tl {#1}
7027 }

```

`\__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。

```

7028 \cs_new_protected:Npn \__xeCJK_visible_space_fallback_auxii:N #1
7029 {
7030     \group_begin:
7031     \exp_args:No \__xeCJK_set_visible_space_size:n
7032     { \dim_use:N \tex_fontdimen:D 2 ~ \tex_font:D }
7033     \cs_new_protected:Npe #1
7034     { \group_begin: \tex_the:D \tex_font:D ~~~~2423 \group_end: }
7035     \group_end:
7036 }
7037 \cs_new_protected:Npn \__xeCJK_set_visible_space_size:n #1
7038 {
7039     \fontencoding { \UnicodeEncodingName }
7040     \tl_set:Nn \f@family { lmtt }
7041     \selectfont
7042     \dim_compare:nNnF {#1} = { \tex_fontdimen:D 2 ~ \tex_font:D }
7043     {
7044         \fontsize
7045         {
7046             \dim_eval:n
7047             {
7048                 \f@size pt *
7049                 \dim_ratio:nn {#1} { \tex_fontdimen:D 2 ~ \tex_font:D }
7050             }
7051         }
7052         { \f@baselineskip }
7053     }
7054 }
7055 }

```

5.16 xeCJK 其他选项

LocalConfig 声明载入本地配置文件的选项。

```

7056 \keys_define:nn { xeCJK / options }
7057 {
7058   LocalConfig .choice: ,
7059   LocalConfig / false .code:n =
7060     { \bool_gset_false:N \g__xeCJK_config_bool } ,
7061   LocalConfig / true .code:n =
7062     {
7063       \bool_gset_true:N \g__xeCJK_config_bool
7064       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
7065     } ,
7066   LocalConfig / unknown .code:n =
7067     {
7068       \bool_gset_true:N \g__xeCJK_config_bool
7069       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
7070     } ,
7071   LocalConfig .default:n = { true }
7072 }
7073 \tl_new:N \g__xeCJK_config_name_tl
7074 \bool_new:N \g__xeCJK_config_bool

```

CJKnumber 和 **indentfirst** 是过时选项。

```

7075 \keys_define:nn { xeCJK / options }
7076 {
7077   CJKnumber .code:n =
7078     { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { CJKnumb } } ,
7079   indentfirst .code:n =
7080     { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { indentfirst } } ,
7081   normalindentfirst .code:n =
7082     { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { } }
7083 }
7084 \__xeCJK_msg_new:nn { option-deprecated }
7085 {
7086   The~`#1'~option~is~deprecated.\\
7087   \tl_if_empty:nF {#2}
7088     { You~may~load~the~package~`#2'~after~xeCJK~to~use~its~function.\\ }
7089 }

```

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

```

7090 \keys_define:nn { xeCJK / options }
7091 {
7092   quiet .code:n =
7093     {
7094       \msg_redirect_module:nnn { xeCJK } { warning } { info }
7095       \msg_redirect_module:nnn { xeCJK } { info } { none }
7096       \xeCJK_if_package_loaded:nF { fontspec }
7097         { \PassOptionsToPackage { quiet } { fontspec } }
7098     } ,
7099   silent .code:n =
7100     {
7101       \msg_redirect_module:nnn { xeCJK } { warning } { none }
7102       \msg_redirect_module:nnn { xeCJK } { info } { none }
7103       \xeCJK_if_package_loaded:nF { fontspec }
7104         { \PassOptionsToPackage { silent } { fontspec } }
7105     } ,
7106   unknown .code:n =
7107     {
7108       \xeCJK_if_package_loaded:nTF { fontspec }
7109         { \__xeCJK_error:ne { key-unknown } { \l_keys_key_str } }
7110         { \PassOptionsToPackage { \l_keys_key_str } { fontspec } }
7111     }
7112 }

```

```

7113 \__xeCJK_msg_new:nn { key-unknown }
7114 {
7115     Sorry,~but~xeCJK/options~does~not~have~a~key~called~`#1'.\\
7116     The~key~`#1'~is~being~ignored.
7117 }

```

5.17 xeCJK 初始化设置

```

\CJKsymbol 7118 \cs_new_eq:NN \CJKsymbol \use:n
\CJKpunctsymbol 7119 \cs_new_eq:NN \CJKpunctsymbol \use:n

```

xeCJK 宏包的初始化设置。

```

7120 \keys_set:nn { xeCJK / options }
7121 {
7122     CJKglue = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
7123     CJKecglue = { ~ } ,
7124     xCJKecglue = false ,
7125     CheckSingle = false ,
7126     PlainEquation = false ,
7127     CheckFullRight = false ,
7128     CJKspace = false ,
7129     CJKmath = false ,
7130     xeCJKactive = true ,
7131     LocalConfig = true ,
7132     LoadFandol = true ,
7133     RubberPunctSkip = true ,
7134     Verb = env ,
7135     EmboldenFactor = 4 ,
7136     SlantFactor = 0.167 ,
7137     PunctStyle = quanjiao ,
7138     NewLineCS = { \par \[ } ,
7139     EnvCS = { \begin \end } ,
7140     WidowPenalty = { 10 000 } ,
7141     NoBreakCS = { \footnote \footnotemark \nobreak } ,
7142     KaiMingPunct = { ~~~~3002 ~~~~ff0e ~~~~ff1f ~~~~ff01 } ,
7143     LongPunct = { ~~~~2014 ~~~~2e3a ~~~~2025 ~~~~2026 } ,
7144     NoBreakLongPunct = { ~~~~2025 ~~~~2026 } ,
7145     MiddlePunct = { ~~~~2013 ~~~~2014 ~~~~2e3a ~~~~2027 ~~~~00b7 ~~~~30fb ~~~~ff65 } ,
7146     AllowBreakBetweenPuncts = false
7147 }
7148 \defaultCJKfontfeatures { Script = CJK }

```

半字线连接号¹⁶应为半角宽度。

```

7149 \xeCJKsetwidth { ~~~~2013 } { 0.5 em }

```

执行宏包选项,并载入 fontspec 宏包。

```

7150 \cs_if_exist:NTF \ProcessKeyOptions
7151 { \ProcessKeyOptions [ xeCJK / options ] }
7152 {
7153     \RequirePackage { l3keys2e }
7154     \ProcessKeysOptions { xeCJK / options }
7155 }
7156 \RequirePackage { fontspec } [ 2020/02/03 ]

```

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

```

7157 \tl_const:Ne \c__xeCJK_encoding_tl { \g_fontspec_encoding_tl }

```

对不能通过 \xeCJKsetup 设置的选项给出警告。

```

7158 \keys_define:nn { xeCJK / options }
7159 {
7160     LocalConfig .code:n =
7161     { \__xeCJK_warning:ne { option-invalid } { \l_keys_key_str } }
7162 }

```

¹⁶见《夹用英文的中文文本的标点符号用法(草案)》5.13 节。

```

7163 \__xeCJK_msg_new:nn { option-invalid }
7164 {
7165   The~`#1'~option~can~only~be~set~in~the~optional~argument~to~the\\
7166   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
7167   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
7168 }

\CJKrmdefault 7169 \tl_if_exist:NF \CJKrmdefault
\JKsfdefault 7170 { \tl_new:N \CJKrmdefault \tl_gset:Nn \CJKrmdefault { rm } }
\JKttdefault 7171 \tl_if_exist:NF \JKsfdefault
\JKfamilydefault 7172 { \tl_new:N \JKsfdefault \tl_gset:Nn \JKsfdefault { sf } }
7173 \tl_if_exist:NF \JKttdefault
7174 { \tl_new:N \JKttdefault \tl_gset:Nn \JKttdefault { tt } }
7175 \tl_new:N \l__xeCJK_family_default_init_tl
7176 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
7177 \tl_set:Nx \l__xeCJK_family_default_init_tl
7178 {
7179   \exp_not:N \__xeCJK_family_default_wrap:n
7180   {
7181     \tl_if_exist:NTF \JKfamilydefault
7182     { \exp_not:o \JKfamilydefault }
7183     { \exp_not:N \CJKrmdefault }
7184   }
7185 }
7186 \tl_if_exist:NF \JKfamilydefault
7187 { \tl_new:N \JKfamilydefault }
7188 \tl_gset_eq:NN \JKfamilydefault \l__xeCJK_family_default_init_tl

```

`\xeCJKsetup` 在导言区或文档中设置 xeCJK 的接口。

```

7189 \NewDocumentCommand \xeCJKsetup { +m }
7190 {
7191   \keys_set:nn { xeCJK / options } {#1}
7192   \tex_ignorespaces:D
7193 }

\xeCJKsetemboldenfactor 7194 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
\xeCJKsetslantfactor 7195 { \xeCJKsetup { EmboldenFactor = {#1} } }
7196 \NewDocumentCommand \xeCJKsetslantfactor { m }
7197 { \xeCJKsetup { SlantFactor = {#1} } }

\punctstyle 7198 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
\xeCJKplainchr 7199 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }

\JKsetecglue 7200 \NewDocumentCommand \JKsetecglue { m } { \xeCJKsetup { JKecglue = {#1} } }
7201 \cs_new_eq:NN \xeCJKsetecglue \JKsetecglue

\JKspace 7202 \NewDocumentCommand \JKspace { } { \xeCJKsetup { JKspace = true } }
\JKnospace 7203 \NewDocumentCommand \JKnospace { } { \xeCJKsetup { JKspace = false } }

\xeCJKallowbreakbetweenpuncts 7204 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
\xeCJKnobreakbetweenpuncts 7205 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
7206 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
7207 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }

\xeCJKenablefallback 7208 \NewDocumentCommand \xeCJKenablefallback { }
\xeCJKdisablefallback 7209 { \xeCJKsetup { AutoFallBack = true } }
7210 \NewDocumentCommand \xeCJKdisablefallback { }
7211 { \xeCJKsetup { AutoFallBack = false } }

\xeCJKsetcharclass 7212 \__xeCJK_msg_new:nn { xeCJKsetcharclass-deprecated }
7213 {
7214   \token_to_str:N \xeCJKsetcharclass\ is~deprecated~and~
7215   does~not~update~CJKmath~char~lists.\\
7216   Use~\token_to_str:N \xeCJKDeclareCharClass\ instead.
7217 }
7218 \NewDocumentCommand \xeCJKsetcharclass { m m m }
7219 {
7220   \__xeCJK_error:n { xeCJKsetcharclass-deprecated }
7221   \xeCJK_set_char_class:nnn {#1} {#2} {#3}
7222   \xeCJKResetPunctClass
7223 }

```

5.18 兼容性修补

`\xeCJK@update@fam` 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。

```

7224 \cs_new_protected:Npn \xeCJK@update@fam
7225 {
7226   \addto@hook \everymath
7227   {
7228     \__xeCJK_update_main_fam:
7229     \__xeCJK_update_block_fam:
7230   }
7231 }
7232 \cs_new_protected:Npn \__xeCJK_update_main_fam:
7233 {
7234   \group_begin:
7235   \xeCJK_select_font:
7236   \exp_last_unbraced:NNNo \group_end:
7237   \tex_textfont:D \c_xeCJK_math_fam_int \tex_the:D \tex_font:D
7238 }
7239 \cs_new_protected:Npn \__xeCJK_update_block_fam:
7240 {
7241   \prop_if_empty:NF \g__xeCJK_block_fam_prop
7242   {
7243     \prop_map_function:NN
7244     \g__xeCJK_block_fam_prop
7245     \__xeCJK_update_block_fam:nn
7246   }
7247 }
7248 \cs_new_protected:Npn \__xeCJK_update_block_fam:nn #1#2
7249 {
7250   \int_set:Nn \l__xeCJK_fam_int {#2}
7251   \group_begin:
7252   \xeCJK_select_font:n {#1}
7253   \exp_last_unbraced:NNNo \group_end:
7254   \tex_textfont:D \l__xeCJK_fam_int \tex_the:D \tex_font:D
7255 }
7256 \__xeCJK_after_end_preamble:n
7257 {
7258   \bool_lazy_and:nnT
7259   { \g__xeCJK_math_bool }
7260   { \cs_if_exist_p:N \Url@MathSetup }
7261   { \tl_put_right:Nn \Url@MathSetup { \xeCJK@update@fam } }
7262 }

```

`\(` 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`)。但是它会创建一个边界, 使 `xeCJK` 不能看到 `\relax` 后面出现的 `$`, 从而不能加入间距¹⁷。使用 $\epsilon\text{-}\text{\TeX}$ 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。

```

7263 \cs_new_protected:Npn \__xeCJK_math_robust:N #1
7264 {
7265   \group_begin: \exp_args:NcNc \group_end:
7266   { \__xeCJK_math_robust_aux:NN } #1 { \cs_to_str:N #1 ~ }

```

¹⁷<http://tex.stackexchange.com/q/124773>

```

7267 }
7268 \cs_new_protected:Npn \__xeCJK_math_robust_aux:NN #1#2
7269 {
7270   \exp_args:Ne \str_case:nnTF { \cs_replacement_spec:N #1 }
7271   {
7272     { \x@protect #1 \protect #2 } { }
7273     { \protect #2 } { }
7274   }
7275   { \__xeCJK_math_robust:NN #1#2 }
7276   { \__xeCJK_math_robust:NN #1#1 }
7277 }
7278 \cs_new_protected:Npn \__xeCJK_math_robust:NN #1#2
7279 {
7280   \str_if_eq:eeTF { \cs_argument_spec:N #2 } { }
7281   {
7282     \exp_args:No \tl_if_head_eq_meaning:nNTF {#2} \scan_stop:
7283     {
7284       \cs_gset_protected:Npe #1
7285       { \tl_tail:N #2 }
7286     }
7287     {
7288       \cs_if_eq:NNTF #1 \ensuremath
7289       {
7290         \cs_gset_protected:Npe #1
7291         { \exp_not:o {#2} }
7292       }
7293       {
7294         \__xeCJK_warning:nee { robust-failure }
7295         { \token_to_str:N #1 } { \token_to_meaning:N #2 }
7296       }
7297     }
7298   }
7299   {
7300     \__xeCJK_warning:nee { robust-failure }
7301     { \token_to_str:N #1 } { \token_to_meaning:N #2 }
7302   }
7303 }
7304 \__xeCJK_msg_new:nnn { robust-failure }
7305 { xeCJK~can~not~make~`#1'~robust. }
7306 {
7307   The~current~meaning~of~`#1'~is:\\
7308   \iow_indent:n {#2}
7309 }
7310 \cs_if_eq:NNTF \(\ \math
7311 {
7312   \__xeCJK_math_robust:N \(\
7313   \cs_set_eq:NN \math \(\
7314 }
7315 {
7316   \__xeCJK_math_robust:N \(\
7317   \__xeCJK_math_robust:N \math
7318 }
7319 \cs_if_eq:NNTF \) \endmath
7320 {
7321   \__xeCJK_math_robust:N \)
7322   \cs_set_eq:NN \endmath \)
7323 }
7324 {
7325   \__xeCJK_math_robust:N \)
7326   \__xeCJK_math_robust:N \endmath
7327 }
7328 \__xeCJK_math_robust:N \ensuremath
7329 % 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
7330 % 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
7331 % 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
7332 % 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 |$|、
7333 % \tn{ } 或 \tn{ensuremath} 时，在 math-on 前报告首类别；结尾是对应公式

```

```

7334 % 时, 把 |math| 记为当前 capture 的确定末类别。 \tn{ensuremath} 已在数学
7335 % 模式中时不重复处理当前列表。
7336 \clist_map_inline:nn { ( , math , ensuremath }
7337 { \__xeCJK_boundary_reserve:n {#1} }
7338 \cs_if_eq:NNTF \ ( \math
7339 {
7340   \cs_new_eq:NN \__xeCJK_orig_inline_math_begin: \ (
7341   \cs_gset_protected:Npn \ (
7342     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_inline_math_begin: }
7343   \cs_gset_eq:NN \math \ (
7344 }
7345 {
7346   \cs_new_eq:NN \__xeCJK_orig_inline_math_begin: \ (
7347   \cs_gset_protected:Npn \ (
7348     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_inline_math_begin: }
7349   \cs_new_eq:NN \__xeCJK_orig_math_begin: \math
7350   \cs_gset_protected:Npn \math
7351     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_math_begin: }
7352 }
7353 \cs_new_eq:NN \__xeCJK_orig_ensuremath: \ensuremath
7354 \cs_gset_protected:Npn \ensuremath
7355 {
7356   \mode_if_math:TF
7357     { \__xeCJK_orig_ensuremath: }
7358     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_ensuremath: }
7359 }
7360 % 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,
7361 % 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,
7362 % 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
7363 % 首尾类别。
7364 \cs_new_protected:Npn \__xeCJK_boundary_register_text_command:N #1
7365 {
7366   \group_begin: \exp_args:NcNc \group_end:
7367   { \__xeCJK_boundary_register_text_command:NN }
7368   #1 { \cs_to_str:N #1 ~ }
7369 }
7370 \cs_new_protected:Npn \__xeCJK_boundary_register_text_command:NN #1#2
7371 {
7372   \__xeCJK_boundary_reserve:N #1
7373   \__xeCJK_boundary_reserve:N #2
7374   \exp_args:Nc \cs_new_eq:NN
7375     { \__xeCJK_boundary_orig_\cs_to_str:N #1 :n } #2
7376   \cs_gset_protected:Npn #2 ##1
7377   {
7378     \__xeCJK_boundary_if_math_edge:nTF {##1}
7379     {
7380       \__xeCJK_boundary_inline_stream_begin:n { auto }
7381       \__xeCJK_boundary_math_begin:n {##1}
7382       \use:c { \__xeCJK_boundary_orig_\cs_to_str:N #1 :n }
7383       { ##1 \__xeCJK_boundary_math_end:n {##1} }
7384       \__xeCJK_boundary_inline_stream_end:n { auto }
7385     }
7386     { \use:c { \__xeCJK_boundary_orig_\cs_to_str:N #1 :n } {##1} }
7387   }
7388 }
7389 \clist_map_inline:nn
7390 {
7391   \textrm , \textsf , \texttt , \textnormal , \textbf , \textmd ,
7392   \textit , \textsl , \textsc , \textup , \emph
7393 }
7394 { \__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

```

7395 \cs_set_eq:NN \xeCJK@family \xeCJK_switch_family:e
7396 \ctex_gadd_ltxhook:nn { rmfamily } { \xeCJK@family { \CJKrmdefault } }
7397 \ctex_gadd_ltxhook:nn { sffamily } { \xeCJK@family { \CJKsfdefault } }

```



```

7398 \ctex_gadd_ltxhook:nn { ttfamily } { \xeCJK@family { \CJKttdefault } }
7399 \ctex_gadd_ltxhook:nn { normalfont } { \xeCJK@family { \CJKfamilydefault } }

7400 <@@=>

```

`\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。

```

7401 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
7402 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
7403 \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。

```

7404 \cs_new_protected:Npn \xeCJK@italiccorr
7405 {
7406   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7407   { \xeCJK_italic_correction: }
7408   { \@@italiccorr }
7409 }

7410 <@@=xeCJK>

```

`\xeCJK_italic_correction:` 修复倾斜校正, 并处理汉字后面的空格。

```

7411 \cs_new_protected:Npn \xeCJK_italic_correction:
7412 { \__xeCJK_if_last_kern:T { \__xeCJK_italic_correction: } }
7413 \cs_new_protected:Npn \__xeCJK_italic_correction:
7414 {
7415   \dim_case:nnF { \tex_lastkern:D }
7416   {
7417     { \__xeCJK_node:n { default } }
7418     {
7419       \xeCJK_remove_node: \tex_italiccorrection:D
7420       \xeCJK_make_node:n { default }
7421     }
7422     { \__xeCJK_node:n { CJK } }
7423     {
7424       \xeCJK_remove_node: \tex_italiccorrection:D
7425       \xeCJK_make_node:n { CJK }
7426       \__xeCJK_italic_correction_aux:
7427     }
7428     { \__xeCJK_node:n { CJK-space } }
7429     {
7430       \xeCJK_remove_node: \tex_italiccorrection:D
7431       \xeCJK_make_node:n { CJK-space }
7432       \__xeCJK_italic_correction_aux:
7433     }
7434   }
7435   { \tex_italiccorrection:D }
7436 }

```

`\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:`)。

```

7437 \cs_new_protected:Npn \__xeCJK_italic_correction_aux:
7438 {
7439   \exp_after:wN \exp_after:wN \exp_after:wN
7440   \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN

```

```

7441 \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7442 \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7443 \xeCJK_ignore_spaces:w
7444 }

```

`\g__xeCJK_xetex_allocator_int` L^AT_EX 2_ε 2015/01/01 接管了 `\newXeTeXintercharclass`。

```

7445 \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。

```

7446 \__xeCJK_after_end_preamble:n
7447 {
7448   \int_compare:nNf
7449   { \c__xeCJK_class_begin_int + \seq_count:N \g__xeCJK_new_class_seq } =
7450   { \g__xeCJK_xetex_allocator_int }
7451   {
7452     \int_step_inline:nnn
7453     { \c__xeCJK_class_begin_int + 1 }
7454     { \g__xeCJK_xetex_allocator_int }
7455     {
7456       \seq_if_in:NnF \g__xeCJK_new_class_seq {#1}
7457       { \__xeCJK_set_others_toks:n {#1} }
7458     }
7459   }
7460 }
7461 \cs_new_protected:Npn \__xeCJK_set_others_toks:n #1
7462 {
7463   \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
7464   \seq_map_inline:Nn \g__xeCJK_CJK_class_seq
7465   {
7466     \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
7467     \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
7468     \xeCJK_app_inter_class_toks:nne {##1} { Others }
7469     { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
7470     \xeCJK_pre_inter_class_toks:nne { Others } {##1}
7471     { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
7472     \tl_if_blank:eT
7473     { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
7474     {
7475       \xeCJK_copy_inter_class_toks:nnnn
7476       { Others } { Boundary } { Default } { Boundary }
7477     }
7478     \tl_if_blank:eT
7479     { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
7480     {
7481       \xeCJK_copy_inter_class_toks:nnnn
7482       { Boundary } { Others } { Boundary } { Default }
7483     }
7484   }
7485 }

```

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

```

\__xeCJK_inactive_group_end:
7486 \cs_new_protected:Npn \__xeCJK_inactive_group_begin:
7487 { \group_begin: \makeXeCJKinactive }
7488 \cs_new_eq:NN \__xeCJK_inactive_group_end: \group_end:

```

`\__xeCJK_patch_text_command:` 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。

```

7489 \prop_const_from_keyval:Nn \c__xeCJK_ambiguous_char_prop
7490 {
7491   "00B7 = \textperiodcentered \textcentereddot \textcdot ,
7492   "2013 = \textendash ,
7493   "2014 = \textemdash ,
7494   "2018 = \textquoteleft \textgrq ,
7495   "2019 = \textquoteright ,

```

```

7496 "201C = \textquotedblleft \textgrqq ,
7497 "201D = \textquotedblright ,
7498 "2025 = \texthdofor ,
7499 "2026 = \textellipsis ,
7500 "2027 = \texthyphenationpoint ,
7501 "2E3A = \texttwoemdash
7502 }
7503 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_text_command: }
7504 \cs_new_protected:Npn \__xeCJK_patch_text_command:
7505 {
7506   \xeCJK_if_package_loaded:nTF { xunicode }
7507   { \__xeCJK_patch_xunicode_ambiguous_char: }
7508   {
7509     \exp_args:Ne \__xeCJK_patch_tuenc_ambiguous_char:n
7510     { \UnicodeEncodingName }
7511     \__xeCJK_patch_tuenc_accent:
7512     \__xeCJK_patch_tuenc_composite:
7513   }
7514 }
7515 \cs_new_protected:Npn \__xeCJK_patch_xunicode_ambiguous_char:
7516 {
7517   \RequirePackage { xunicode-addon }
7518   \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
7519   {
7520     \tl_map_inline:nn { ##2 }
7521     {
7522       \xunadd_set_begin_hook:nn { ####1 }
7523       { \__xeCJK_inactive_group_begin: }
7524       \xunadd_set_end_hook:nn { ####1 }
7525       { \__xeCJK_inactive_group_end: }
7526     }
7527   }
7528   \xunadd_append_begin_hook:n { \xeCJK_make_boundary: }
7529 }
7530 \cs_new_protected:Npn \__xeCJK_patch_tuenc_ambiguous_char:n #1
7531 {
7532   \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
7533   {
7534     \tl_map_inline:nn { ##2 }
7535     {
7536       \cs_if_exist:NF ####1
7537       { \DeclareTextSymbol ####1 {#1} { ##1 } }
7538       \__xeCJK_patch_ambiguous_char:nN {#1} ####1
7539     }
7540   }
7541 }
7542 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nN #1#2
7543 {
7544   \exp_args:Ne \__xeCJK_patch_ambiguous_char:nn
7545   { #1 \token_to_str:N #2 }
7546   { #1 - #2 }
7547 }
7548 \cs_new_protected:Npe \__xeCJK_patch_ambiguous_char:nNn #1#2#3
7549 {
7550   \exp_not:N \exp_args:Ne
7551   \exp_not:N \__xeCJK_patch_ambiguous_char:nn
7552   {
7553     \c_backslash_str #1
7554     \exp_not:N \token_to_str:N #2 -
7555     \exp_not:N \token_to_str:N #3
7556   }
7557   { #1 - #2#3 }
7558 }
7559 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nn #1#2
7560 {
7561   \cs_if_free:cF {#1}
7562   { \exp_args:Nc \__xeCJK_patch_ambiguous_char:Nn {#1} {#2} }

```

```

7563 }
7564 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:Nn #1#2
7565 {
7566   \token_if_chardef:NTF #1
7567   {
7568     \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
7569     { \int_eval:n {#1} }
7570     \cs_set_protected:Npe #1
7571     { \__xeCJK_ambiguous_char:n { \tex_Uchar:D #1 } }
7572   }
7573   {
7574     \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
7575     { \int_eval:n { \exp_after:wN ` #1 } }
7576     \cs_set_protected:Npe #1
7577     { \__xeCJK_ambiguous_char:n { \exp_not:o {#1} } }
7578   }
7579 }
7580 \cs_new_protected:Npn \__xeCJK_ambiguous_char:n #1
7581 {
7582   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7583   { \__xeCJK_inactive_group_begin: #1 \__xeCJK_inactive_group_end: }
7584   {#1}
7585 }
7586 \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

```

7587 \cs_new_protected:Npn \xeCJK_text_composite_patch:
7588 {
7589   \str_if_eq:eeT { \f@encoding } { \UnicodeEncodingName }
7590   { \xeCJK_make_boundary: }
7591 }

```

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。

```

7592 \cs_new_protected:Npn \__xeCJK_patch_tuenc_composite:
7593 {
7594   \cs_set_nopar:Npn \@text@composite@x
7595   {
7596     \xeCJK_text_composite_patch:
7597     \cs_if_exist_use:NF
7598   }
7599 }

```

`\__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。

```

7600 \group_begin:
7601 \char_set_catcode_other:n { "A0 }
7602 \cs_new_protected:Npn \__xeCJK_patch_tuenc_accent:
7603 {
7604   \cs_set_protected_nopar:Npn \add@unicode@accent ##1 ##2
7605   {
7606     \tl_if_blank:nTF { ##2 } { ^^a0 } { ##2 }
7607     \tex_Uchar:D \tex_numexpr:D ##1 \scan_stop:
7608   }
7609 }
7610 \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{\cyrahbha}{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。

```

7611 \prop_const_from_keyval:Nn \c__xeCJK_middle_dot_prop
7612 {
7613   T2A = \cyrchvcrs ,
7614   T2B = \cyrchldsc ,
7615   T2C = \cyrahbha ,
7616   X2  = \cyrchldsc ,
7617   TS1 = \textperiodcentered ,
7618   LY1 = \textperiodcentered ,
7619   T1  = \r u ,
7620   T4  = \B t ,
7621   T5  = \` \ecircumflex
7622 }
7623 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_middle_dot: }
7624 \cs_new_protected:Npn \__xeCJK_patch_middle_dot:
7625 {
7626   \prop_map_inline:Nn \c__xeCJK_middle_dot_prop
7627   { \__xeCJK_patch_middle_dot:nw { ##1 } ##2 \q_stop }
7628   \__xeCJK_patch_ambiguous_char:nNn { T5 } \` { ^ - e }
7629 }
7630 \cs_new_protected:Npn \__xeCJK_patch_middle_dot:nw #1#2#3 \q_stop
7631 {
7632   \tl_if_empty:nTF {#3}
7633   { \__xeCJK_patch_ambiguous_char:nN {#1} #2 }
7634   { \__xeCJK_patch_ambiguous_char:nNn {#1} #2 {#3} }
7635 }

```

pifont 宏包的符号\ding{183} 也有冲突。

```

7636 \__xeCJK_package_hook:nn { pifont }
7637 {
7638   \RenewDocumentCommand \Pifont { m }
7639   { \mode_leave_vertical: \makexeCJKinactive \usefont { U } {#1} { m } { n } }
7640 }

```

\xeCJKchar \char 是 TeX 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 mtpro2)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 LuaTeX-jan 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 LuaTeX-jan 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。

```

7641 \cs_new_protected:Npn \xeCJK_default_char:w
7642 {
7643   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7644   {
7645     \c_group_begin_token
7646     \tex_afterassignment:D \__xeCJK_default_char_aux:
7647     \l__xeCJK_default_char_int =
7648   }
7649   { \tex_char:D }

```

```

7650 }
7651 \cs_new_protected:Npn \__xeCJK_default_char_aux:
7652 {
7653     \tex_XeTeXinterchartokenstate:D = \c_zero_int
7654     \tex_char:D \l__xeCJK_default_char_int
7655     \c_group_end_token
7656 }
7657 \int_new:N \l__xeCJK_default_char_int
7658 \cs_new_eq:NN \xeCJKchar \xeCJK_default_char:w

```

mtpro2 的 `\overcbrace` 和 `\undercbrace` 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。

```

7659 \__xeCJK_package_hook:nn { mtpo2 }
7660 {
7661     \cs_set_eq:NN \__xeCJK_mtpo_overcbrace:w \overcbrace
7662     \cs_gset_protected:Npn \overcbrace #1
7663     {
7664         \makexeCJKinactive
7665         \__xeCJK_mtpo_overcbrace:w {#1}
7666     }
7667     \cs_set_eq:NN \__xeCJK_mtpo_undercbrace:w \undercbrace
7668     \cs_gset_protected:Npn \undercbrace #1
7669     {
7670         \makexeCJKinactive
7671         \__xeCJK_mtpo_undercbrace:w {#1}
7672     }
7673 }

```

`\__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:
\__xeCJK_save_um_char:
7674 \__xeCJK_package_hook:nn { unicode-math }
7675 {
7676     \prop_const_from_keyval:Nn \c__xeCJK_um_ambiguous_char_prop
7677     {
7678         "00B7 = \cdotp ,
7679         "2025 = \enleadertwodots ,
7680         "2026 = \unicodeellipsis
7681     }
7682     \cs_new_protected:Npn \__xeCJK_save_um_char:
7683     {
7684         \cs_set_protected:Npe \__xeCJK_restore_um_char:
7685         {
7686             \prop_map_function:NN
7687             \c__xeCJK_um_ambiguous_char_prop
7688             \__xeCJK_restore_um_char_aux:nn
7689         }
7690     }
7691     \cs_new_eq:NN \__xeCJK_restore_um_char: \prg_do_nothing:
7692     \cs_new:Npn \__xeCJK_restore_um_char_aux:nn #1#2
7693     {
7694         \__xeCJK_gset_mathcodenum:nn
7695         { \int_value:w #1 }
7696         { \int_value:w \tex_Umathcodenum:D #1 }
7697     }
7698     \cs_new_protected:Npn \__xeCJK_gset_mathcodenum:nn #1#2
7699     {
7700         \int_compare:nNnF { \tex_Umathcodenum:D #1 } = {#2}
7701         { \tex_global:D \tex_Umathcodenum:D #1 = #2 ~ }
7702     }
7703 }

```

`\__xeCJK_patch_microtype_get_slot:` 兼容 microtype。

```

7704 \cs_new_protected:Npn \__xeCJK_patch_microtype_get_slot:

```

```

7705 {
7706   \cs_new_eq:NN \xeCJK@original@get@slot \MT@get@slot@
7707   \cs_set_eq:NN \MT@get@slot@ \xeCJK@microtype@get@slot
7708   \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
7709 }
7710 \cs_new_protected_nopar:Npn \xeCJK@microtype@get@slot
7711 {
7712   \int_compare:nNtT \MT@char < \c_zero_int
7713   { \__xeCJK_get_ambiguous_slot: }
7714   \xeCJK@original@get@slot
7715 }
7716 \cs_new_protected:Npn \__xeCJK_get_ambiguous_slot:
7717 {
7718   \prop_get:NnT \g__xeCJK_ambiguous_slot_prop
7719   { \MT@encoding - \tex_the:D \MT@toks } \l__xeCJK_tmp_tl
7720   { \cs_set_eq:NN \MT@char \l__xeCJK_tmp_tl }
7721 }
7722 \cs_new_protected:Npn \xeCJK@microtype@restore@pickupfont
7723 { \__xeCJK_gadd_font_initial_hook:n { \MT@ltx@pickupfont } }
7724 \__xeCJK_package_hook:nn { microtype }
7725 {
7726   \cs_if_free:NF \MT@get@slot@
7727   { \__xeCJK_patch_microtype_get_slot: }
7728   \MT@addto@setup { \xeCJK@microtype@restore@pickupfont }
7729 }

```

简单处理与 `hyperref` 宏包的兼容问题。

`hyperref` 的行内锚点会在 `xeCJK` 的 `marker` 与后续字符之间插入不可见节点, 使 `Boundary→{Default,CJK}` 路径的 `marker` 探测失败, 锚点右侧的 `\CJKecglue` 因此丢失(#1047)。这类命令没有可见输出, 故与 `hypdoc` 的 `\HD@target` 一样注册为 `transparent`。

下面已覆盖三个出口。区分依据是调用点, 不是锚点内容:

- 驱动层的 `\hyper@anchor` 承接经 `\hyper@@anchor` 进来的锚点。`\hypertarget` 的两个分支最终都走这里——`\@hyper@@anchor` 在 `\ifHy@activeanchor` 为假时统一调用 `\hyper@anchor`, 与目标内容是否为空无关。它直接排出裸的 `pdf:dest` 什么的。
- `\Hy@raisedlink` 承接需要抬升的锚点: 无编号标题(`\section*`、`\chapter*` 以及目录、参考文献等自动生成的无编号标题)、`caption`、公式编号、脚注、`\bibitem`, 以及下游手工包裹的写法, 例如 `ctxdoc` 的 `\exptarget` 定义为 `\Hy@raisedlink{\hypertarget{<name>}}`。它在水平模式下输出 `\penalty\@M` 加一个 `\smash` 后的 `hbox(0+0)x0` 来盛放 `pdf:dest special`。注意目录条目不走这条路: `\contentsline` 用 `\hyper@linkstart`/`\hyper@linkend` 做链接, 与抬升锚点无关。

因此两处都要注册: 只注册 `\Hy@raisedlink` 时所有 `\hypertarget` 仍缺间距, 只注册 `\hyper@anchor` 时 `\exptarget` 一类写法与无编号标题锚点仍缺间距。

第三个出口是 `\__hyp_target_raise:n: \phantomsection` 与 `\MakeLinkTarget` 走它, 编号标题的锚点也经过它。它自己排出同构的 `\penalty\@M` 加 `\smash` 抬升盒子, 不经过 `\Hy@raisedlink`。这个函数不接受通用命令 `hook` (LaTeX hook 机制拒绝 `expl3` 私有函数), 因此用 `\__xeCJK_boundary_wrap_transparent_onearg_braced:NN` 包装——必须是带花括号转发参数的那个变体, 理由见该函数的说明。

已知未覆盖: `\hyper@anchorstart` 的裸调用。`\pdfbookmark` 直接写 `\hyper@anchorstart{<name>}` 既不经 `\hyper@@anchor` 也不经两个抬升出口(计数器实测三者均为 0), 因此右侧仍会丢失一枚 `\CJKecglue` (实测 38.33002pt 对 oracle 41.66002pt)。同类裸调用在 `hyperref` 与各驱动里还有若干处。

两种就手的补法都已实测不可行: 注册 `\@pdfm@dest(\hyper@anchor` 与 `\hyper@anchorstart` 的共同下游) 会让盒宽暴涨并报出十余处错误, 因为它的参数含有待展开内容; 注册 `\hyper@anchorstart` 本身不报错, 但也不生效——`\pdfbookmark` 仍为 38.33002pt, 而已覆

盖的三处不受影响。为什么 transparent 在这个入口上无效尚未查明, 这条路径需要单独设计适配器, 另立议题跟踪。

判断这类问题应当读分派函数的分支并用计数器实测, 不能按公开命令的参数形态推测走哪条路, 也不能因为「注册若干处之后报告的现象消失」就断定出口的数量。本节的机制陈述先后被独立复核推翻四次:「按目标是否为空分派」「只有两个出口」「包装失败源于 begin 钩子的赋值」「只有三个出口」, 每一次都是从「注册这些之后现象消失」推出更强的断言, 每一次都由一支计数器探针或一次隔离实验推翻。因此本节只列已覆盖与已知未覆盖两份清单, 不再给出总数。

\hyper@anchor 由驱动定义, hxdetex、hluatex、hpdftex 与 hdvipdfm 使用同一名称, hdvips 经 pdfmark 定义同名命令; 名称虽然一致, 各驱动的定义时机仍由 hyperref 决定, 注册前先检查是否存在, 避免装出无效 hook。hyperref 在 \AtEndOfPackage 阶段载入驱动, 故包尾钩子里的存在性检查时机正确。

\Hy@raisedlink 这一侧不能用 post-transparent: 那个 after-only 变体只在命令结束后搬移末尾零尺寸盒子下方的 marker 与候选 glue, 而 \Hy@raisedlink 在盒子之前还排出了 \penalty, marker 与盒子并不相邻, 搬移条件不成立。transparent 在入口就取走 marker 与可选源码空格, 锚点节点排出后再原样恢复, 因而两种节点次序都能覆盖。

\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。

```

7730 \cs_new_protected:Npn \__xeCJK_boundary_register_hyperref_annot:
7731 {
7732   \clist_map_inline:nn
7733   {
7734     Hy@BeginAnnot , Hy@EndAnnot , hyper@link , hyper@linkurl ,
7735     href , hyperref
7736   }
7737   { \__xeCJK_boundary_reserve:n {##1} }
7738   \cs_gset_eq:NN \__xeCJK_orig_Hy_BeginAnnot:n \Hy@BeginAnnot
7739   \cs_gset_eq:NN \__xeCJK_orig_Hy_EndAnnot: \Hy@EndAnnot
7740   \cs_gset_protected:Npn \Hy@BeginAnnot ##1
7741   {
7742     \__xeCJK_boundary_inline_stream_begin:n { auto }
7743     \__xeCJK_orig_Hy_BeginAnnot:n { ##1 }
7744   }
7745   \cs_gset_protected:Npn \Hy@EndAnnot
7746   {
7747     \int_compare:nNtT { \c@Hy@AnnotLevel } = { 1 }
7748     {
7749       \__xeCJK_if_last_math:T
7750       {
7751         \tl_if_eq:cnF
7752         {
7753           g__xeCJK_boundary_capture_
7754           \int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl
7755         }
7756         { math }
7757         { \__xeCJK_boundary_capture_class:n { default } }
7758       }
7759     }
7760     \__xeCJK_orig_Hy_EndAnnot:
7761     \__xeCJK_boundary_inline_stream_end:n { auto }
7762   }
7763   \cs_gset_eq:NN \__xeCJK_orig_hyper_link:nnn \hyper@link
7764   \cs_gset_protected:Npn \hyper@link ##1##2##3

```



```

7765 {
7766   \_xeCJK_orig_hyper_link:nnn {##1} {##2}
7767   {
7768     \_xeCJK_boundary_math_begin:n {##3}
7769     ##3
7770     \_xeCJK_boundary_math_end:n {##3}
7771   }
7772 }
7773 \cs_gset_eq:NN \_xeCJK_orig_hyper_linkurl:nn \hyper@linkurl
7774 \cs_gset_protected:Npn \hyper@linkurl ##1##2
7775 {
7776   \_xeCJK_orig_hyper_linkurl:nn
7777   {
7778     \_xeCJK_boundary_math_begin:n {##1}
7779     ##1
7780     \_xeCJK_boundary_math_end:n {##1}
7781   }
7782   {##2}
7783 }
7784 \_xeCJK_boundary_register_transparent:n { Hy@raisedlink }
7785 \cs_if_exist:NT \hyper@anchor
7786 { \_xeCJK_boundary_register_transparent:n { hyper@anchor } }
7787 \cs_if_exist:NT \_hyp_target_raise:n
7788 {
7789   \_xeCJK_boundary_wrap_transparent_onearg_braced:NN
7790   \_xeCJK_orig_hyp_target_raise:n \_hyp_target_raise:n
7791 }
7792 }
7793 \_xeCJK_package_hook:nn { hyperref }
7794 {
7795   \pdfstringdefDisableCommands
7796   {
7797     \_xeCJK_gobble_CJKfamily:
7798     \xeCJK_cs_clear:N \_xeCJK_inactive_group_begin:
7799     \xeCJK_cs_clear:N \_xeCJK_inactive_group_end:
7800     \xeCJK_cs_clear:N \makexeCJKinactive
7801     \xeCJK_cs_clear:N \xeCJK_text_composite_patch:
7802   }
7803   \_xeCJK_boundary_register_hyperref_annot:
7804 }

```

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。

```

7805 % \tn{colorbox} 和 \tn{fcolorbox} 的扫描器都会把可见正文留作
7806 % \tn{color@b@x} 的第三个参数。先包装这个共同入口, 再安装 wrapped-box
7807 % 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。
7808 % \pkg{xcolor} 会重定义这个私有函数, 因此它的 package hook 需要按
7809 % 当前定义重新保存原函数。
7810 \cs_new_protected:Npn \_xeCJK_boundary_color_box:nnn #1#2#3
7811 {
7812   \_xeCJK_boundary_math_begin:n {#3}
7813   \_xeCJK_orig_color_box:nnn {#1} {#2}
7814   { #3 \_xeCJK_boundary_math_end:n {#3} }
7815 }
7816 \cs_new_protected:Npn \_xeCJK_boundary_prepare_color_box:
7817 {
7818   \_xeCJK_boundary_reserve:n { color@b@x }
7819   \cs_if_eq:NNF \color@b@x \_xeCJK_boundary_color_box:nnn
7820   {
7821     \cs_gset_eq:NN \_xeCJK_orig_color_box:nnn \color@b@x
7822     \cs_gset_eq:NN \color@b@x \_xeCJK_boundary_color_box:nnn

```

```

7823     }
7824 }
7825 \cs_new_protected:Npn \__xeCJK_boundary_textcolor:nnn #1#2#3
7826 {
7827     \__xeCJK_boundary_if_math_edge:nTF {#3}
7828     {
7829         \__xeCJK_boundary_inline_stream_begin:n { auto }
7830         \__xeCJK_boundary_math_begin:n {#3}
7831         \__xeCJK_orig_textcolor:nnn {#1} {#2}
7832         { #3 \__xeCJK_boundary_math_end:n {#3} }
7833         \__xeCJK_boundary_inline_stream_end:n { auto }
7834     }
7835     { \__xeCJK_orig_textcolor:nnn {#1} {#2} {#3} }
7836 }
7837 \cs_new_protected:Npn \__xeCJK_boundary_register_textcolor:
7838 {
7839     \__xeCJK_boundary_reserve:n { @textcolor }
7840     \__xeCJK_boundary_reserve:n { textcolor }
7841     \cs_if_eq:NNF \@textcolor \__xeCJK_boundary_textcolor:nnn
7842     {
7843         \cs_gset_eq:NN \__xeCJK_orig_textcolor:nnn \@textcolor
7844         \cs_gset_eq:NN \@textcolor \__xeCJK_boundary_textcolor:nnn
7845     }
7846 }
7847 \cs_new_protected:Npn \__xeCJK_boundary_register_color:
7848 {
7849     \__xeCJK_boundary_reserve:n { colorbox }
7850     \__xeCJK_boundary_reserve:n { fcolorbox }
7851     \__xeCJK_boundary_register_transparent:n { set@color }
7852     \__xeCJK_boundary_register_transparent:n { reset@color }
7853     \__xeCJK_boundary_prepare_color_box:
7854     \__xeCJK_boundary_register_wrapped_box:nn { color@b@x } { auto }
7855     \__xeCJK_boundary_register_textcolor:
7856 }
7857 \__xeCJK_package_hook:nn { color }
7858 { \__xeCJK_boundary_register_color: }
7859 \__xeCJK_package_hook:nn { xcolor }
7860 { \__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。

```

7861 \cs_new_protected:Npn \__xeCJK_boundary_register_l3color:
7862 {
7863     \__xeCJK_boundary_wrap_transparent_onearg:NN
7864     \__xeCJK_orig_color_select:N \__color_select:N
7865     \__xeCJK_boundary_wrap_transparent_noarg:NN
7866     \__xeCJK_orig_color_backend_reset: \__color_backend_reset:
7867 }
7868 \__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 / \__xeCJK_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。

```

7869 \__xeCJK_package_hook:nn { hypdoc }
7870 { \__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, 不经过 `\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。

```
7871 \cs_new_protected:Npn \__xeCJK_boundary_register_siunitx:
7872 {
7873   \__xeCJK_boundary_register_stream:nn { unit } { default }
7874   \__xeCJK_boundary_register_stream:nn { qty } { default }
7875   \__xeCJK_boundary_register_stream:nn { num } { default }
7876   \cs_if_exist:NT \si
7877     { \__xeCJK_boundary_register_stream:nn { si } { default } }
7878   \cs_if_exist:NT \SI
7879     { \__xeCJK_boundary_register_stream:nn { SI } { default } }
7880 }
7881 \__xeCJK_package_hook:nn { siunitx }
7882 { \__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 $\text{\XeTeX}$ 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。

```
7883 \cs_new_protected:Npn \__xeCJK_boundary_register_url:
7884 {
7885   \clist_map_inline:nn { url , path , nolinkurl , \Url@z }
7886     { \__xeCJK_boundary_reserve:n {##1} }
7887   \cs_if_exist:NF \__xeCJK_orig_Url_z:n
7888     {
7889       \cs_set_eq:NN \__xeCJK_orig_Url_z:n \Url@z
7890       \cs_gset_protected:Npn \Url@z ##1
7891         {
7892           \__xeCJK_boundary_inline_stream_begin:n { default }
7893           \__xeCJK_orig_Url_z:n {##1}
7894           \__xeCJK_boundary_inline_stream_end:n { default }
7895         }
7896     }
7897 }
7898 \__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→{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; `\shortvrb` 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 在它之后重放实际未类别, 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。

```
7899 \cs_new_protected:Npn \_xeCJK_flush_language_whatsit:
7900 { \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: 7901 \cs_new_protected:Npn \_xeCJK_boundary_register_verb:
7902 {
7903   \cs_if_exist:NF \_xeCJK_orig_verb:
7904   {
7905     \cs_set_eq:NN \_xeCJK_orig_verb: \verb
7906     \cs_gset_protected:Npn \verb
7907     {
7908       \_xeCJK_boundary_inline_stream_begin:n { auto }
7909       \_xeCJK_orig_verb:
7910     }
7911     \tl_gput_right:Nn \verb@egroup
7912     {
7913       \_xeCJK_flush_language_whatsit:
7914       \_xeCJK_boundary_inline_stream_end:n { auto }
7915     }
7916   }
7917 }
7918 \_xeCJK_boundary_reserve:n { verb }
7919 \_xeCJK_boundary_reserve:n { verb@egroup }
7920 \_xeCJK_after_preamble:n { \_xeCJK_boundary_register_verb: }
```

l3doc 类用 _codedoc_meta:n 排版 \meta 命令的参数: 内部由 _codedoc_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 \CJKecglue, 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 _codedoc_meta:n 的参数装进 \hbox:n。hbox 隔离了内部的 X_YTeX 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 \CJKecglue。

仅用 hbox 包装会让入口的 math 节点遮住左侧 marker。适配器因此还要用固定 Default 首尾的 stream capture 包围整条命令: framework 负责外侧两端, hbox 只负责尖括号内部的输出规范化。

capture 必须包住 \texttt 而不只是内层参数排版函数: _xeCJK_boundary_capture_-begin: 在入口处把 \CJKecglue 排入临时盒并读成 skip 数值, 缓存的度量因此取决于 capture 开始时生效的字体。l3doc 的 \meta 定义为 \texttt{_codedoc_meta:n {<参数>}}, 若只包围内层的 _codedoc_meta:n, capture 就落在 \texttt 已经切到等宽字体之后: 左边界重放的是等宽字体的 \CJKecglue (lm 下为不可伸缩的 5.25pt), 而右边界在 capture 结束、字体恢

复之后求值, 得到正文字体的弹性 3.33pt, 两侧因此不对称(#1046)。把 capture 上移到公开 \meta 之后, 两端都在同一字体上下文中求值。

\Arg、\marg、\oarg、\parg 不需要这层 capture: 它们在 __codedoc_meta:n 两侧各自排出等宽的 {、[、(等实字符, 本身就构成正常的 CJK→Default 边界, 去掉内层 capture 前后的宽度与可见排版结果完全相同(各类上下文实测), 节点列表上少了内层 capture 原本留下的一对零效果 default marker kern (± 0.0002 pt)。这些入口仍写入保留表, 防止用户接口把通用 hook 叠加到共享的内部实现上。

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 __codedoc_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: 7921 \cs_new_protected:Npn \__xeCJK_boundary_register_codedoc_meta:
7922 {
7923   \cs_if_exist:NTF \__codedoc_meta:n
7924   {
7925     \clist_map_inline:nn
7926     { \__codedoc_meta:n , Arg , marg , oarg , par }
7927     { \__xeCJK_boundary_reserve:n {##1} }
7928     \cs_if_exist:NF \__xeCJK_orig_codedoc_meta:n
7929     {
7930       \cs_set_eq:NN \__xeCJK_orig_codedoc_meta:n \__codedoc_meta:n
7931       \cs_gset_protected:Npn \__codedoc_meta:n ##1
7932       { \__xeCJK_orig_codedoc_meta:n { \hbox:n {##1} } }
7933       \__xeCJK_boundary_register_stream:nn { meta } { default }
7934     }
7935   }
7936   {
7937     \bool_lazy_and:nnT
7938     { \cs_if_exist_p:c { ver@doc.sty } }
7939     { \cs_if_free_p:N \__xeCJK_orig_codedoc_meta:n }
7940     {
7941       \__xeCJK_boundary_reserve:n { meta }
7942       \cs_set_eq:NN \__xeCJK_orig_codedoc_meta:n \meta
7943       \DeclareDocumentCommand \meta { m }
7944       {
7945         \__xeCJK_boundary_inline_stream_begin:n { default }
7946         \__xeCJK_orig_codedoc_meta:n { \hbox:n {##1} }
7947         \__xeCJK_boundary_inline_stream_end:n { default }
7948       }
7949     }
7950   }
7951 }
7952 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_register_codedoc_meta: }
7953 % \changes{v3.10.4}{2026/07/20}{将 \tn{eqref} 注册为固定 Default
7954 % 首尾、将 \tn{cs} 注册为仅固定 Default 首端的 stream capture;
7955 % codedoc/doc 的 meta 内部适配器直接包围固定 Default stream, 删除专用
7956 % drain 并覆盖其所有公共调用方 (\#992)。}
7957 % \changes{v3.10.4}{2026/07/20}{将内核 \tn{@setref} (或 \pkg{hyperref}
7958 % 保存的 \tn{real@setref}) 注册为 auto stream capture, 以统一框架取代
7959 % \#991 的专用 saved-node/replay 补丁。}
7960 % \tn{eqref} 的圆括号和 \tn{meta} 的数学尖括号决定了命令两端
7961 % 始终是 Default; 参数内部即使包含中文, 也不能把外侧 oracle
7962 % 误判为裸中文。 \tn{cs} 只有开头的反斜线固定为 Default, 末尾仍由
7963 % 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
7964 % 真实末类别。
7965 \__xeCJK_at_end_preamble:n
7966 {
7967   \cs_if_exist:NTF \real@setref
7968   { \__xeCJK_boundary_register_stream:nn { real@setref } { auto } }
7969   { \__xeCJK_boundary_register_stream:nn { @setref } { auto } }
7970   \cs_if_exist:NT \eqref
7971   { \__xeCJK_boundary_register_stream:nn { eqref } { default } }
7972   \cs_if_exist:NT \cs
7973   { \__xeCJK_boundary_register_stream:nn { cs } { first-default } }
```

```
7974 }
```

`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: 7975 \cs_new_protected:Npn \__xeCJK_boundary_register_biblatex_pagetracker:
7976 {
7977   \cs_if_exist:NT \blx@pagetracker
7978     { \__xeCJK_boundary_register_transparent:n { \blx@pagetracker } }
7979 }
7980 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_register_biblatex_pagetracker: }
7981 % 用户声明在全部内建 preamble 未注册之后应用; 普通命令 hook 仍由 LaTeX
7982 % 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,
7983 % 避免不存在的控制序列被 LaTeX 静默跳过。
7984 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_user_register_apply: }
7985 \__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。

```
7986 \__xeCJK_package_hook:nn { cprotect }
7987 {
7988   \cs_if_free:NF \icprotect
7989     { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
7990 }
```

在 `listings` 宏包后自动载入 `xeCJK-listings`。

```
7991 \__xeCJK_package_hook:nn { listings }
7992 { \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。

```
7993 \__xeCJK_package_hook:nn { everyysel }
7994 {
7995   \cs_if_exist:NF \@EverySelectfont@Legacy
7996     { \cs_undefine:c { ver@CJK . \c__xeCJK_package_ext_tl } }
7997 }
```

`\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。

```
7998 \ctex_at_begin_package:nn { CJKnumb }
7999 {
8000   \tl_new:N \l__xeCJK_CJK_version_tl
8001   \tl_set_eq:Nc \l__xeCJK_CJK_version_tl { ver@CJK . \c__xeCJK_package_ext_tl }
8002   \tl_set:cn { ver@CJK . \c__xeCJK_package_ext_tl } { 9999/99/99 }
```

```

8003 \cs_new_protected:Npn \CJKaddEncHook #1#2
8004 {
8005   \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
8006   {
8007     \group_begin:
8008     \cs_set_eq:NN \Unicode \xeCJK_unicode_char:nn
8009     \cs_set_eq:NN \def \xdef
8010     #2
8011     \group_end:
8012     \str_gset:Nn \CJK@tenthousand { ~~~~4e07 }
8013     \str_gset:Nn \CJK@hundredmillion { ~~~~4ebf }
8014     \tl_if_exist:NF \CJK@UnicodeEnc
8015     { \tl_const:Nn \CJK@UnicodeEnc { UTF8 } }
8016     \cs_if_exist:NF \Unicode
8017     { \cs_new_eq:NN \Unicode \xeCJK_unicode_char:nn }
8018   }
8019 }
8020 \cs_new:Npn \xeCJK_unicode_char:nn #1#2
8021 { \tex_Uchar:D \tex_numexpr:D (#1) * 256 + (#2) \scan_stop: }
8022 }
8023 \ctex_at_end_package:nn { CJKnumb }
8024 { \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。

```

8025 \bool_if:NT \g__xeCJK_config_bool
8026 {
8027   \ExplSyntaxOff
8028   \file_input:n { \g__xeCJK_config_name_tl .cfg }
8029   \ExplSyntaxOn
8030 }
8031 </package>

```

5.19 xeCJKfntef

```

8032 <*/fntef>
8033 \PassOptionsToPackage { normalem } { ulem }
8034 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
8035 \ProcessOptions \scan_stop:
8036 \RequirePackage { xeCJK }
8037 \RequirePackage { ulem }
8038 \RequirePackage { l3draw }
8039 \RequirePackage { accsupp }
8040 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }
\xeCJK_hook_for_ulem: 8041 \cs_new_protected:Npn \xeCJK_hook_for_ulem:
8042 {
8043   \xeCJK_ulem_detect_node:
8044   \l__xeCJK_ulem_text_format_tl
8045   \bool_if:NF \l__xeCJK_ulem_hook_used_bool
8046   {
8047     \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
8048     \__xeCJK_ulem_hook:
8049   }
8050   \xeCJK_ulem_begin_node:
8051 }
8052 \cs_new_protected:Npn \__xeCJK_ulem_hook:
8053 {
8054   \__xeCJK_ulem_initial:
8055   \bool_if:NTF \l__xeCJK_ulem_periodic_bool
8056   {
8057     \seq_gput_right:Nn \g__xeCJK_ulem_periodic_later_seq { 0 }
8058     \cs_set_eq:NN \UL@spfactor \__xeCJK_ulem_spfactor_pending:
8059     \xeCJK_swap_cs:NN
8060     \UL@leaders \xeCJK_ulem_periodic_leaders:

```

```

8061 \cs_set_eq:NN \__xeCJK_ulem_periodic_var_leaders:
8062 \xeCJK_ulem_periodic_var_leaders:
8063 \cs_set_eq:NN \xeCJK_ulem_right_skip:
8064 \__xeCJK_ulem_periodic_right_skip:
8065 }
8066 {
8067 \bool_if:NT \l__xeCJK_ulem_subtract_bool
8068 {
8069 \xeCJK_swap_cs:NN \UL@leaders \xeCJK_ulem_leaders:
8070 \cs_set_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:
8071 \cs_set_eq:NN \xeCJK_ulem_right_skip: \__xeCJK_ulem_right_skip:
8072 }
8073 }
8074 \bool_if:NT \l__xeCJK_ulem_hidden_bool
8075 { \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_hidden_box: }
8076 \bool_if:NTF \l__xeCJK_ulem_skip_bool
8077 {
8078 \cs_set_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
8079 \cs_set_eq:NN \__xeCJK_ulem_hskip_aux:n \xeCJK_ulem_hskip:n
8080 }
8081 {
8082 \xeCJK_swap_cs:NN \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
8083 \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
8084 \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
8085 }
8086 \xeCJK_glue_to_skip:nN
8087 {
8088 \cs_set_eq:NN \ \tex_space:D
8089 \cs_set_eq:NN \penalty \tex_penalty:D
8090 \cs_set_eq:NN \hskip \skip_horizontal:N
8091 \CJKglue
8092 } \l__xeCJK_ccglue_skip
8093 \xeCJK_glue_to_skip:nN
8094 {
8095 \cs_set_eq:NN \ \tex_space:D
8096 \cs_set_eq:NN \penalty \tex_penalty:D
8097 \cs_set_eq:NN \hskip \skip_horizontal:N
8098 \CJKecglue
8099 } \l__xeCJK_ecglue_skip
8100 \xeCJK_glue_to_skip:nN { \xeCJK_space_glue: } \l__xeCJK_space_skip
8101 \cs_set_protected:Npn \CJKglue
8102 { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
8103 \cs_set_protected:Npn \CJKecglue
8104 { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
8105 \cs_set_protected:Npn \xeCJK_space_glue:
8106 { \__xeCJK_ulem_glue:n \l__xeCJK_space_skip }
8107 \cs_set_eq:NN \xeCJK_punct_node:N \use_none:n
8108 \cs_set_eq:NN \xeCJK_if_last_punct:TF \use_ii:nn
8109 \keys_set:nn { xeCJK / options }
8110 { CheckFullRight = false , xCJKecglue = false }
8111 }
8112 \skip_new:N \l__xeCJK_space_skip
8113 \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

```

8114 \clist_map_inline:nn
8115 {
8116 \UL@word , \ULon , \UL@on , \UL@onin ,
8117 \uline , \uuline , \uwave , \sout , \xout , \dashuline , \dotuline ,
8118 \xeCJKfntefon , \CJKunderline , \CJKunderwave , \CJKunderdblline ,
8119 \CJKsout , \CJKxout , \CJKunderanyline , \CJKunderansymbol , \CJKunderdot
8120 }
8121 { \__xeCJK_boundary_reserve:n {#1} }
8122 \cs_new_protected:Npn \xeCJK_ulem_word:nw #1 ~
8123 {
8124 \__xeCJK_ulem_start:w #1 ~
8125 \exp_after:wN \if_meaning:w \exp_after:wN \UL@end #1

```



```

8126     \exp_after:wN \__xeCJK_ulem_end:
8127   \else:
8128     \exp_after:wN \__xeCJK_ulem_loop:nw
8129   \fi:
8130 }
8131 \cs_new_protected:Npn \__xeCJK_ulem_end:
8132 {
8133   \c_group_end_token
8134   \c_group_end_token
8135   \tex_unskip:D \tex_unskip:D \tex_unskip:D
8136   \xeCJK_ulem_right_skip:
8137   \xeCJK_ulem_group_end:
8138   \xeCJK_ulem_right_node:
8139   \__xeCJK_boundary_inline_stream_end:n { auto }
8140   \bool_set_false:N \l__xeCJK_ulem_stream_started_bool
8141   \int_set:Nn \tex_spacefactor:D { \UL@spfactor }
8142 }
8143 \cs_new_protected:Npn \__xeCJK_ulem_loop:nw
8144 {
8145   \reverse_if:N \if_mode_math:
8146   \reverse_if:N \if_dim:w \tex_lastskip:D = \c_zero_dim
8147   \skip_gset_eq:NN \UL@skip \tex_lastskip:D
8148   \tex_unskip:D
8149   \UL@stop \UL@leaders
8150   \fi:
8151   \fi:
8152   \xeCJK_ulem_word:nw \prg_do_nothing:
8153 }
8154 \cs_new_protected:Npn \__xeCJK_ulem_start:w
8155 { \exp_after:wN \UL@start }
8156 \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:

```

8157 \cs_new_protected:Npn \xeCJK_ulem_left:
8158 {
8159   \xeCJK_ulem_left_node:
8160   \xeCJK_make_group_tag:
8161 }
8162 \cs_new_eq:NN \xeCJK_ulem_left_node: \prg_do_nothing:
8163 \cs_new_protected:Npn \xeCJK_ulem_detect_node:
8164 {
8165   \scan_stop:
8166   \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
8167   {
8168     \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
8169     \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
8170     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8171   }
8172   {
8173     \dim_set_eq:NN \l__xeCJK_tmp_dim \tex_lastkern:D
8174     \tex_unkern:D
8175     \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_tmp_dim }
8176     {
8177       \tex_unkern:D
8178       \cs_set_protected:Npe \xeCJK_ulem_left_node:
8179       {
8180         \tex_kern:D - \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
8181         \tex_kern:D \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
8182       }
8183       \cs_set_protected:Npn \xeCJK_ulem_begin_node:
8184       { { \xeCJK_make_node:n { ulem-begin } } }
8185       \cs_set_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
8186     }
8187     {
8188       \tex_kern:D \l__xeCJK_tmp_dim
8189       \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
8190       \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:

```

```

8191         \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8192     }
8193 }
8194 }
8195 \xeCJK_declare_node:n { ulem-begin }
8196 \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 `\CJKe glue` 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。

```

8197 \cs_new_protected:Npn \__xeCJK_ulem_hskip_first:n #1
8198 {
8199     \xeCJK_if_last_node:nTF { ulem-begin }
8200     {
8201         \xeCJK_remove_node:
8202         \skip_horizontal:n {#1}
8203     }
8204     { \xeCJK_ulem_hskip:n {#1} }
8205     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8206 }
8207 \cs_new_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
8208 \cs_new_protected:Npn \xeCJK_ulem_hskip:n #1
8209 { { \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。

```

8210 \cs_new_protected:Npn \xeCJK_ulem_right:
8211 {
8212     \scan_stop:
8213     \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
8214     { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
8215     {
8216         \dim_compare:nNnTF \tex_lastkern:D = { 3sp }
8217         { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
8218         {
8219             \exp_args:NNo \tex_unkern:D
8220             \__xeCJK_ulem_right_aux:n { \dim_use:N \tex_lastkern:D }
8221         }
8222     }
8223 }
8224 \cs_new_protected:Npn \__xeCJK_ulem_right_aux:n #1
8225 {
8226     \dim_compare:nNnTF \tex_lastkern:D = { - #1 }
8227     {
8228         \tex_unkern:D
8229         \cs_gset_protected:Npn \xeCJK_ulem_right_node:
8230         {
8231             \tex_kern:D - #1 \exp_stop_f:
8232             \tex_kern:D #1 \exp_stop_f:
8233         }
8234         \tl_gset:Nx \UL@spfactor { \int_use:N \tex_spacefactor:D }
8235     }
8236     {
8237         \tex_kern:D #1 \exp_stop_f:
8238         \xeCJK_cs_gclear:N \xeCJK_ulem_right_node:
8239     }
8240 }
8241 \cs_new_eq:NN \xeCJK_ulem_right_node: \prg_do_nothing:

```

`\xeCJK_ulem_var_leaders:` ulem 通常把每段装饰向左右各扩展 `\UL@pixel`; 对带减号形式, 第一次画装饰时改为先向右
 移动 `\UL@pixel`, 使装饰从正文内部开始。后续片段仍使用原来的 `\UL@leaders`, 所以正文和
 中文字间距共享普通 `\leaders` 的相位。

```

8242 \cs_new_protected:Npn \xeCJK_ulem_leaders:
8243 { \__xeCJK_ulem_var_leaders: }
8244 \cs_new_protected:Npn \xeCJK_ulem_var_leaders:

```

```

8245 {
8246   \scan_stop:
8247   \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8248   {
8249     \skip_horizontal:N \UL@pixel
8250     \UL@leadtype \skip_horizontal:N \UL@skip
8251     \skip_horizontal:n { - \UL@pixel }
8252     \cs_gset_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_leaders:
8253   }
8254 }
8255 \cs_new_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:

```

\xeCJK_ulem_periodic_leaders: 普通 \leaders 只放置完整的装饰盒。它能让同一行中的各个片段共享相位, 但首尾可见余量会随命令的水平位置变化。周期装饰因此只在命令的首段和末段增加 PDF 裁切: 底下的 \xeCJK_ulem_periodic_var_leaders: \xeCJK_ulem_periodic_first_leaders: \xeCJK_ulem_periodic_later_leaders: \xeCJK_ulem_periodic_clipped_leaders:nnn 区间向两侧各扩展一个完整周期, 再把可见范围精确限制到正文的对称边界。内部正文片段和 \CJKglue 仍使用原来的普通 leaders, 不改变伸缩和断行。

参数 #2,#3 是相对正文左右边界向内移动的距离。普通形式的首末段都使用 -\UL@pixel,-\UL@pixel 即左右各外伸半个周期。带减号形式的首段使用 \UL@pixel,\UL@pixel; 真正的末段使用 -\UL@pixel,\UL@pixel, 把内部接点接上, 同时让命令右端仍内缩半个周期。首段之后的第一个中文字间距在断点两侧各放一个净宽为零的半周期连接片段; 不换行时两半相接, 换行时则分别止于上一行末尾和下一行开头。单片段命令只使用首段裁切。若正文没有产生节点, 第一次调用 leaders 的内容其实只是 ulem 追加的结尾语法空格。此时保留原来的三段 glue 结构, 让 ulem 随后用三次 \unskip 删除, 不能把这枚空格改造成含 PDF special 和 kern 的裁切结构。正文中的真实空格出现在 \UL@spfactor 哨兵更新之前, 不会被误认为结尾语法空格。

```

8256 \cs_new_protected:Npn \xeCJK_ulem_periodic_leaders:
8257 { \__xeCJK_ulem_periodic_var_leaders: }
8258 \cs_new_protected:Npn \xeCJK_ulem_periodic_var_leaders:
8259 {
8260   \scan_stop:
8261   \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8262   {
8263     \cs_if_eq:NNTF \UL@spfactor \__xeCJK_ulem_spfactor_pending:
8264     { \__xeCJK_ulem_periodic_first_leaders: }
8265     {
8266       \skip_if_eq:nnTF { \UL@skip } { \box_wd:N \UL@box }
8267       { \__xeCJK_ulem_periodic_first_leaders: }
8268       { \xeCJK_ulem_periodic_leaders: }
8269     }
8270   }
8271 }
8272 \cs_new_protected:Npn \__xeCJK_ulem_periodic_first_leaders:
8273 {
8274   \bool_if:NNTF \l__xeCJK_ulem_subtract_bool
8275   {
8276     \__xeCJK_ulem_periodic_clipped_leaders:nnn
8277     { \UL@skip } { \UL@pixel } { \UL@pixel }
8278   }
8279   {
8280     \__xeCJK_ulem_periodic_clipped_leaders:nnn
8281     { \UL@skip } { - \UL@pixel } { - \UL@pixel }
8282   }
8283   \cs_gset_eq:NN \__xeCJK_ulem_periodic_var_leaders:
8284   \__xeCJK_ulem_periodic_later_leaders:
8285 }
8286 \cs_new_eq:NN \__xeCJK_ulem_periodic_var_leaders:
8287 \xeCJK_ulem_periodic_var_leaders:
8288 \cs_new_protected:Npn \__xeCJK_ulem_periodic_later_leaders:
8289 {
8290   \scan_stop:
8291   \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8292   {
8293     \cs_if_eq:NNTF \UL@spfactor \__xeCJK_ulem_spfactor_pending:

```

```

8294     {
8295         \seq_get_right:NN
8296         \g__xeCJK_ulem_periodic_later_seq \l__xeCJK_tmp_tl
8297         \tl_if_eq:NnTF \l__xeCJK_tmp_tl { 1 }
8298         { \xeCJK_ulem_periodic_leaders: }
8299         {
8300             \seq_gpop_right:NN
8301             \g__xeCJK_ulem_periodic_later_seq \l__xeCJK_tmp_tl
8302             \seq_gput_right:Nn \g__xeCJK_ulem_periodic_later_seq { 1 }
8303             \bool_if:NTF \l__xeCJK_ulem_subtract_bool
8304             {
8305                 \__xeCJK_ulem_periodic_clipped_leaders:nnn
8306                 { \c_zero_skip } { - \UL@pixel } { \c_zero_dim }
8307                 \xeCJK_ulem_periodic_leaders:
8308                 \__xeCJK_ulem_periodic_clipped_leaders:nnn
8309                 { \c_zero_skip } { \c_zero_dim } { - \UL@pixel }
8310             }
8311             { \xeCJK_ulem_periodic_leaders: }
8312         }
8313     }
8314     { \xeCJK_ulem_periodic_leaders: }
8315 }
8316 }
8317 \cs_new_protected:Npn \__xeCJK_ulem_periodic_clipped_leaders:nnn #1#2#3
8318 {
8319     \skip_set:Nn \l__xeCJK_tmp_skip { #1 - #2 - #3 }
8320     \dim_compare:nNnTF { \l__xeCJK_tmp_skip } > { \c_zero_dim }
8321     {
8322         \tex_kern:D \dim_eval:n {#2}
8323         \tex_special:D { x:gsave }
8324         \tex_special:D
8325         {
8326             pdf:literal~
8327             0~
8328             \dim_to_decimal_in_bp:n { - \box_dp:N \ULC@box - 1pt } ~
8329             \dim_to_decimal_in_bp:n { \l__xeCJK_tmp_skip } ~
8330             \dim_to_decimal_in_bp:n
8331             { \box_ht:N \ULC@box + \box_dp:N \ULC@box + 2pt } ~
8332             re~W~n
8333         }
8334         \tex_kern:D \dim_eval:n { - 2 \UL@pixel }
8335         \xeCJK_no_break:
8336         \UL@leadtype
8337         \skip_horizontal:n { \l__xeCJK_tmp_skip + 4 \UL@pixel }
8338         \tex_kern:D \dim_eval:n { - 2 \UL@pixel }
8339         \tex_special:D { x:grestore }
8340         \tex_kern:D \dim_eval:n {#3}
8341     }
8342     { \skip_horizontal:n {#1} }
8343 }
8344 \bool_new:N \l__xeCJK_ulem_periodic_bool
8345 \seq_new:N \g__xeCJK_ulem_periodic_later_seq
8346 \cs_new_eq:NN \__xeCJK_ulem_spfactor_pending: \scan_stop:

```

`\xeCJK_ulem_right_skip:` 在装饰完全画好之后检查末尾节点。带减号形式把最后一段的装饰宽度减少两个 `\UL@pixel`，再补回一个等宽空白，使装饰停在正文右边界以内 `\UL@pixel`，同时保持命令和正文的宽度不变。若末段窄到不足以容纳缩减后的装饰，则只保留两端空白，不构造负宽度的 leaders。

```

8347 \cs_new_eq:NN \xeCJK_ulem_right_skip: \prg_do_nothing:
8348 \cs_new_protected:Npn \__xeCJK_ulem_right_skip:
8349 {
8350     \int_case:nn { \tex_lastnodetype:D }
8351     {
8352         { \c__xeCJK_hlist_node } { \__xeCJK_ulem_right_skip_hbox: }
8353         { \c__xeCJK_glue_node } { \__xeCJK_ulem_right_skip_glue: }
8354         { \c__xeCJK_penalty_node } { \__xeCJK_ulem_right_skip_penalty: }
8355     }

```

```

8356 }
8357 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_hbox:
8358 {
8359   \box_set_to_last:N \l__xeCJK_tmp_box
8360   \__xeCJK_if_last_kern:TF
8361     { \__xeCJK_ulem_right_skip_kern: }
8362     { \__xeCJK_ulem_right_skip_glue: }
8363   \box_use_drop:N \l__xeCJK_tmp_box
8364 }
8365 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_kern:
8366 {
8367   \dim_set:Nn \l__xeCJK_tmp_dim { - \box_wd:N \l__xeCJK_tmp_box }
8368   \dim_compare:nNnT \tex_lastkern:D = \l__xeCJK_tmp_dim
8369     {
8370       \tex_unkern:D
8371       \__xeCJK_ulem_right_skip_glue:
8372       \tex_kern:D \l__xeCJK_tmp_dim
8373     }
8374 }
8375 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_glue:
8376 {
8377   \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8378     {
8379       \tex_unskip:D
8380       \skip_set:Nn \l__xeCJK_tmp_skip
8381         { \tex_lastskip:D - 2 \UL@pixel }
8382       \tex_unskip:D
8383       \dim_compare:nNnT { \l__xeCJK_tmp_skip } > { \c_zero_dim }
8384         { \UL@leadtype \skip_horizontal:N \l__xeCJK_tmp_skip }
8385       \skip_horizontal:N \UL@pixel
8386     }
8387 }
8388 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_penalty:
8389 {
8390   \int_set_eq:NN \l__xeCJK_tmp_int \tex_lastpenalty:D
8391   \tex_unpenalty:D
8392   \__xeCJK_if_last_hlist:T
8393     { \__xeCJK_ulem_right_skip_hbox: }
8394   \tex_penalty:D \l__xeCJK_tmp_int
8395 }

```

`\__xeCJK_ulem_periodic_right_skip:` 若命令只有一个装饰片段, 首段函数已经同时裁定左右边界; 出现后续片段时, 末段仍是 `ulem` 原来的负间距、`leaders`、负间距结构。这里取回末段正文宽度, 并用同一裁切函数重画, 使命令右边界不再依赖完整图案盒的取整。每个周期装饰在开始时向全局序列压入自己的状态, 结束时再弹出。这里不能共用一个全局布尔值: 内层周期装饰会在外层完成之前改写它, 导致外层末段根据内层的状态决定是否重画。

```

8396 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip:
8397 {
8398   \seq_gpop_right:NN \g__xeCJK_ulem_periodic_later_seq \l__xeCJK_tmp_tl
8399   \tl_if_eq:NnTF \l__xeCJK_tmp_tl { 1 }
8400     { \__xeCJK_ulem_periodic_right_skip_aux: }
8401     { \prg_do_nothing: }
8402 }
8403 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_aux:
8404 {
8405   \int_case:nn { \tex_lastnodetype:D }
8406     {
8407       { \c__xeCJK_hlist_node } { \__xeCJK_ulem_periodic_right_skip_hbox: }
8408       { \c__xeCJK_glue_node } { \__xeCJK_ulem_periodic_right_skip_glue: }
8409       { \c__xeCJK_penalty_node } { \__xeCJK_ulem_periodic_right_skip_penalty: }
8410     }
8411 }
8412 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_hbox:
8413 {
8414   \box_set_to_last:N \l__xeCJK_tmp_box

```

```

8415 \__xeCJK_if_last_kern:TF
8416 { \__xeCJK_ulem_periodic_right_skip_kern: }
8417 { \__xeCJK_ulem_periodic_right_skip_glue: }
8418 \box_use_drop:N \l__xeCJK_tmp_box
8419 }
8420 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_kern:
8421 {
8422   \dim_set:Nn \l__xeCJK_tmp_dim { - \box_wd:N \l__xeCJK_tmp_box }
8423   \dim_compare:nNtT \tex_lastkern:D = \l__xeCJK_tmp_dim
8424   {
8425     \tex_unkern:D
8426     \__xeCJK_ulem_periodic_right_skip_glue:
8427     \tex_kern:D \l__xeCJK_tmp_dim
8428   }
8429 }
8430 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_glue:
8431 {
8432   \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8433   {
8434     \tex_unskip:D
8435     \skip_set:Nn \l__xeCJK_tmp_skip { \tex_lastskip:D - 2 \UL@pixel }
8436     \tex_unskip:D
8437     \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8438     {
8439       \tex_unskip:D
8440       \bool_if:NTF \l__xeCJK_ulem_subtract_bool
8441       {
8442         \__xeCJK_ulem_periodic_clipped_leaders:nnn
8443         { \l__xeCJK_tmp_skip } { - \UL@pixel } { \UL@pixel }
8444       }
8445       {
8446         \__xeCJK_ulem_periodic_clipped_leaders:nnn
8447         { \l__xeCJK_tmp_skip } { - \UL@pixel } { - \UL@pixel }
8448       }
8449     }
8450   }
8451 }
8452 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_penalty:
8453 {
8454   \int_set_eq:NN \l__xeCJK_tmp_int \tex_lastpenalty:D
8455   \tex_unpenalty:D
8456   \__xeCJK_if_last_hlist:T
8457   { \__xeCJK_ulem_periodic_right_skip_hbox: }
8458   \tex_penalty:D \l__xeCJK_tmp_int
8459 }

```

__xeCJK_ulem_hidden_box: 只画线, 不输出盒子。

```

8460 \cs_new_protected:Npn \__xeCJK_ulem_hidden_box:
8461 {
8462   \tl_if_empty:NF \UL@start
8463   {
8464     \box_set_ht:Nn \l__xeCJK_hidden_box { \box_ht:N \UL@box }
8465     \box_set_dp:Nn \l__xeCJK_hidden_box { \box_dp:N \UL@box }
8466     \box_use:N \l__xeCJK_hidden_box
8467     \xeCJK_no_break:
8468     \xeCJK_ulem_hskip:n { \box_wd:N \UL@box }
8469     \box_use:N \l__xeCJK_hidden_box
8470   }
8471 }
8472 \box_new:N \l__xeCJK_hidden_box
8473 \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:

```

8474 \cs_new_protected:Npn \__xeCJK_ulem_skip_punct_begin:
8475 {

```

```

8476 \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_skip_putbox:
8477 \cs_set_eq:NN \xeCJK_ulem_hskip:n \skip_horizontal:n
8478 }
8479 \cs_new_protected:Npn \__xeCJK_ulem_skip_punct_end:
8480 {
8481 \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_putbox:
8482 \cs_set_eq:NN \xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_aux:n
8483 }
8484 \cs_new_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
8485 \cs_new_protected:Npn \__xeCJK_ulem_skip_putbox:
8486 {
8487 \tl_if_empty:NF \UL@start
8488 { \box_use_drop:N \UL@box }
8489 }

```

__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。

```

8490 \cs_new_protected:Npn \__xeCJK_ulem_initial:
8491 {
8492 \__xeCJK_ulem_swap_cs:NN
8493 \xeCJK_FullLeft_and_Default: \__xeCJK_ulem_FullLeft_and_Default:
8494 \xeCJK_FullLeft_and_CJK: \__xeCJK_ulem_FullLeft_and_CJK:
8495 \xeCJK_FullLeft_and_Boundary: \__xeCJK_ulem_FullLeft_and_Boundary:
8496 \xeCJK_FullRight_and_Default: \__xeCJK_ulem_FullRight_and_Default:
8497 \xeCJK_FullRight_and_CJK: \__xeCJK_ulem_FullRight_and_CJK:
8498 \xeCJK_FullRight_and_CJStarter: \__xeCJK_ulem_FullRight_and_CJStarter:
8499 \xeCJK_FullRight_and_Boundary: \__xeCJK_ulem_FullRight_and_Boundary:
8500 \xeCJK_CJK_and_CJK:N \__xeCJK_ulem_CJK_and_CJK:N
8501 \xeCJK_CJK_and_Boundary:w \__xeCJK_ulem_CJK_and_Boundary:w
8502 \xeCJK@fix@penalty \__xeCJK_ulem_fix_penalty:
8503 \__xeCJK_punct_breakable_kern:n \__xeCJK_ulem_punct_breakable_kern:n
8504 \__xeCJK_Default_and_FullLeft_glue:N \__xeCJK_ulem_Default_and_FullLeft_glue:N
8505 \__xeCJK_Default_and_FullRight_glue:N \__xeCJK_ulem_Default_and_FullRight_glue:N
8506 \__xeCJK_CJK_and_FullLeft_glue:N \__xeCJK_ulem_CJK_and_FullLeft_glue:N
8507 \__xeCJK_CJK_and_FullRight_glue:N \__xeCJK_ulem_CJK_and_FullRight_glue:N
8508 \__xeCJK_Boundary_and_FullLeft_glue:N \__xeCJK_ulem_Boundary_and_FullLeft_glue:N
8509 \q_recursion_tail \q_nil \q_recursion_stop
8510 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
8511 {
8512 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
8513 {
8514 \str_if_eq:nnTF {##1} {####1}
8515 {
8516 \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
8517 { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
8518 \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
8519 { \__xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
8520 }
8521 {
8522 \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
8523 { \__xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
8524 }
8525 }
8526 }
8527 }
8528 \cs_new_protected:Npn \__xeCJK_ulem_swap_cs:NN #1#2
8529 {
8530 \quark_if_recursion_tail_stop:N #1
8531 \xeCJK_swap_cs:NN #1#2
8532 \__xeCJK_ulem_swap_cs:NN
8533 }

```

\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。

```

8534 \cs_new:Npn \xeCJK_if_ulem_patch:TF

```

```

8535 {
8536   \if_meaning:w \ \LA@space
8537   \exp_after:wN \use_ii:nn
8538   \else:
8539   \exp_after:wN \use_i:nn
8540   \fi:
8541 }

\__xeCJK_ulem_CJK_and_Boundary:w 8542 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_Boundary:w
8543 {
8544   \xeCJK_if_ulem_patch:TF
8545   {
8546     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
8547     { \xeCJK_class_group_end: \CJKeclglue }
8548     {
8549       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
8550       { \__xeCJK_ulem_peek_math:w }
8551       { \__xeCJK_ulem_group_end:n { CJK } }
8552     }
8553   }
8554   { \__xeCJK_ulem_CJK_and_Boundary:w }
8555 }
8556 \cs_new_protected:Npn \__xeCJK_ulem_group_end:n #1
8557 {
8558   \xeCJK_class_group_end: \UL@stop
8559   \UL@start { \xeCJK_make_node:n {#1} }
8560   \xeCJK_make_group_tag:
8561   \__xeCJK_boundary_enable_source_space:n {#1}
8562 }

```

__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。

```

8563 \cs_new_protected:Npn \__xeCJK_ulem_peek_math:w
8564 {
8565   \cs_set_eq:NN \__xeCJK_ulem_start:w \__xeCJK_ulem_exp_stop:w
8566   \exp_after:wN \peek_after:Nw
8567   \exp_after:wN \__xeCJK_ulem_peek_math_branches:w
8568   \exp:w \exp_end_continue_f:w
8569 }
8570 \cs_new_protected:Npn \__xeCJK_ulem_peek_math_branches:w
8571 {
8572   \token_if_math_toggle:NTF \l_peek_token
8573   { \xeCJK_class_group_end: \CJKeclglue }
8574   { \__xeCJK_ulem_group_end:n { CJK-space } }
8575 }
8576 \cs_new_protected:Npn \__xeCJK_ulem_exp_stop:w
8577 {
8578   \cs_if_eq:NNTF \UL@start \@empty
8579   { \exp_after:wN \exp_stop_f: }
8580   { \exp_after:wN \UL@start }
8581 }

\__xeCJK_ulem_fix_penalty: 8582 \cs_new_protected:Npn \__xeCJK_ulem_fix_penalty:
8583 {
8584   \xeCJK_if_ulem_patch:TF
8585   { \fix@penalty }
8586   { \__xeCJK_ulem_fix_penalty: }
8587 }

```

__xeCJK_ulem_save_font_state: 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。

```

\__xeCJK_ulem_save_font_state: 8588 \tl_new:N \g__xeCJK_ulem_saved_series_tl
\__xeCJK_ulem_restore_font_state: 8589 \tl_new:N \g__xeCJK_ulem_saved_shape_tl
\g__xeCJK_ulem_saved_series_tl 8590 \tl_new:N \g__xeCJK_ulem_saved_CJK_family_tl
\g__xeCJK_ulem_saved_shape_tl
\g__xeCJK_ulem_saved_CJK_fam_tl

```

¹⁸<https://github.com/CTeX-org/ctex-kit/issues/614>


```

8591 \tl_new:N \g__xeCJK_ulem_saved_CJK_fam_tl
8592 \cs_new_protected:Npn \__xeCJK_ulem_save_font_state:
8593 {
8594   \tl_gset:Ne \g__xeCJK_ulem_saved_series_tl { \f@series }
8595   \tl_gset:Ne \g__xeCJK_ulem_saved_shape_tl { \f@shape }
8596   \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_family_tl \l_xeCJK_family_tl
8597   \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_fam_tl \CJK@family
8598 }
8599 \cs_new_protected:Npn \__xeCJK_ulem_restore_font_state:
8600 {
8601   \tl_set:Ne \f@series { \g__xeCJK_ulem_saved_series_tl }
8602   \tl_set:Ne \f@shape { \g__xeCJK_ulem_saved_shape_tl }
8603   \tl_set_eq:NN \l_xeCJK_family_tl \g__xeCJK_ulem_saved_CJK_family_tl
8604   \tl_set_eq:NN \CJK@family \g__xeCJK_ulem_saved_CJK_fam_tl
8605 }

\__xeCJK_ulem_CJK_and_CJK:N 8606 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_CJK:N
8607 {
8608   \xeCJK_if_ulem_patch:TF
8609   {
8610     \__xeCJK_ulem_save_font_state:
8611     \xeCJK_class_group_end:
8612     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
8613     \__xeCJK_ulem_class_group_begin:
8614     \__xeCJK_ulem_restore_font_state:
8615     \xeCJK_select_font:
8616     \xeCJK_fallback_symbol:NN
8617     \CJKsymbol
8618   }
8619   { \__xeCJK_ulem_CJK_and_CJK:N }
8620 }

\__xeCJK_ulem_class_group_begin: 8621 \cs_new_protected:Npn \__xeCJK_ulem_class_group_begin:
8622 {
8623   \xeCJK_class_group_begin:
8624   \xeCJK_clear_Boundary_and_CJK_toks:
8625 }

\__xeCJK_ulem_between_CJK_blocks:nnN 8626 \cs_new_protected:Npn \__xeCJK_ulem_between_CJK_blocks:nnN #1#2
8627 {
8628   \xeCJK_if_ulem_patch:TF
8629   {
8630     \__xeCJK_ulem_save_font_state:
8631     \xeCJK_class_group_end:
8632     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
8633     \xeCJK_class_group_begin:
8634     \xeCJK_clear_Boundary_and_CJK_toks:
8635     \__xeCJK_ulem_restore_font_state:
8636     \__xeCJK_switch_font:nn {#1} {#2}
8637     \xeCJK_fallback_symbol:NN
8638     \CJKsymbol
8639   }
8640   {
8641     \skip_horizontal:N \l__xeCJK_ccglue_skip
8642     \__xeCJK_switch_font:nn {#1} {#2}
8643     \xeCJK_fallback_symbol:NN
8644     \CJKsymbol
8645   }
8646 }

\__xeCJK_ulem_Default_and_FullLeft_glue:N 8647 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
8648 {
8649   \xeCJK_if_ulem_patch:TF
8650   {
8651     \UL@stop
8652     \__xeCJK_ulem_skip_punct_begin:
8653     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8654     \UL@start

```

```

8655     }
8656     { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
8657 }

```

```

\__xeCJK_ulem_Boundary_and_FullLeft_glue:N 8658 \cs_new_protected:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
8659 {
8660     \xeCJK_if_ulem_patch:TF
8661     {
8662         \UL@stop
8663         \__xeCJK_ulem_skip_punct_begin:
8664         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8665         \UL@start
8666     }
8667     { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
8668 }

```

```

\__xeCJK_ulem_CJK_and_FullLeft_glue:N 8669 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
8670 {
8671     \xeCJK_if_ulem_patch:TF
8672     {
8673         \xeCJK_class_group_end:
8674         \UL@stop
8675         \__xeCJK_ulem_skip_punct_begin:
8676         \__xeCJK_ulem_punct_ccglue:
8677         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8678         \UL@start
8679         \__xeCJK_ulem_class_group_begin:
8680         \xeCJK_select_punct_font:
8681     }
8682     { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
8683 }

```

```

\__xeCJK_ulem_Default_and_FullRight_glue:N 8684 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
8685 {
8686     \xeCJK_if_ulem_patch:TF
8687     {
8688         \UL@stop
8689         \__xeCJK_ulem_skip_punct_begin:
8690         \__xeCJK_punct_if_long:NTF #1
8691         { \xeCJK_allow_break: }
8692         { \xeCJK_no_break: }
8693         \__xeCJK_punct_if_middle:NT #1
8694         {
8695             \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
8696             \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
8697         }
8698         \UL@start
8699     }
8700     { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
8701 }

```

```

\__xeCJK_ulem_CJK_and_FullRight_glue:N 8702 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
8703 {
8704     \xeCJK_if_ulem_patch:TF
8705     {
8706         \xeCJK_class_group_end:
8707         \UL@stop
8708         \__xeCJK_ulem_skip_punct_begin:
8709         \__xeCJK_punct_if_long:NTF #1
8710         { \xeCJK_allow_break: }
8711         { \xeCJK_no_break: }
8712         \__xeCJK_punct_if_middle:NT #1
8713         {
8714             \__xeCJK_ulem_punct_ccglue:
8715             \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
8716             \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
8717         }
8718         \UL@start

```

```

8719     \__xeCJK_ulem_class_group_begin:
8720     \xeCJK_select_punct_font:
8721 }
8722 { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
8723 }

\__xeCJK_ulem_FullLeft_and_Default: 8724 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_Default:
8725 {
8726     \xeCJK_if_ulem_patch:TF
8727     {
8728         \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
8729         {
8730             \xeCJK_get_punct_bounds:No \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8731             \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8732             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
8733             \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8734         }
8735         { \xeCJK_class_group_end: \UL@stop }
8736         \__xeCJK_ulem_skip_punct_end:
8737         \xeCJK_no_break:
8738         \UL@start
8739     }
8740     { \__xeCJK_ulem_FullLeft_and_Default: }
8741 }

\__xeCJK_ulem_FullLeft_and_Boundary: 8742 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_Boundary:
8743 {
8744     \xeCJK_if_ulem_patch:TF
8745     {
8746         \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
8747         {
8748             \xeCJK_get_punct_bounds:No \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8749             \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8750             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
8751             \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8752         }
8753         { \xeCJK_class_group_end: \UL@stop }
8754         \__xeCJK_ulem_skip_punct_end:
8755         \xeCJK_no_break:
8756         \UL@start
8757         \tex_ignorespaces:D
8758     }
8759     { \__xeCJK_ulem_FullLeft_and_Boundary: }
8760 }

\__xeCJK_ulem_FullLeft_and_CJK: 8761 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_CJK:
8762 {
8763     \xeCJK_if_ulem_patch:TF
8764     {
8765         \xeCJK_FullLeft_and_Default:
8766         \__xeCJK_ulem_class_group_begin:
8767         \xeCJK_select_font:
8768     }
8769     { \__xeCJK_ulem_FullLeft_and_CJK: }
8770 }

\__xeCJK_ulem_FullRight_and_Default: 8771 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Default:
8772 {
8773     \xeCJK_if_ulem_patch:TF
8774     {
8775         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8776         \xeCJK_class_group_end:
8777         \UL@stop
8778         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8779         \__xeCJK_ulem_skip_punct_end:
8780         \UL@start
8781     }
8782     { \__xeCJK_ulem_FullRight_and_Default: }
8783 }

```

```

\__xeCJK_ulem_FullRight_and_Boundary: 8784 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Boundary:
8785 {
8786   \xeCJK_if_ulem_patch:TF
8787   {
8788     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8789     \xeCJK_class_group_end:
8790     \UL@stop
8791     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8792     \__xeCJK_ulem_skip_punct_end:
8793     \UL@start
8794     \tex_ignorespaces:D
8795   }
8796   { \__xeCJK_ulem_FullRight_and_Boundary: }
8797 }

\__xeCJK_ulem_FullRight_and_CJK: 8798 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJK:
8799 {
8800   \xeCJK_if_ulem_patch:TF
8801   {
8802     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8803     \xeCJK_class_group_end:
8804     \UL@stop
8805     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8806     \__xeCJK_ulem_punct_ccglue:
8807     \__xeCJK_ulem_skip_punct_end:
8808     \UL@start
8809     \__xeCJK_ulem_class_group_begin:
8810     \xeCJK_select_font:
8811   }
8812   { \__xeCJK_ulem_FullRight_and_CJK: }
8813 }

```

__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。

```

8814 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJStarter:
8815 {
8816   \xeCJK_if_ulem_patch:TF
8817   {
8818     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8819     \xeCJK_class_group_end:
8820     \UL@stop
8821     \xeCJK_no_break:
8822     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8823     \__xeCJK_ulem_punct_ccglue:
8824     \__xeCJK_ulem_skip_punct_end:
8825     \UL@start
8826     \__xeCJK_ulem_class_group_begin:
8827     \xeCJK_select_font:
8828   }
8829   { \__xeCJK_ulem_FullRight_and_CJStarter: }
8830 }

\__xeCJK_ulem_punct_hskip:n 8831 \cs_new_protected:Npn \__xeCJK_ulem_punct_hskip:n
8832 {
8833   \xeCJK_if_ulem_patch:TF
8834   { \xeCJK_ulem_hskip:n }
8835   { \__xeCJK_ulem_punct_hskip:n }
8836 }

\__xeCJK_ulem_punct_breakable_kern:n 8837 \cs_new_protected:Npn \__xeCJK_ulem_punct_breakable_kern:n #1
8838 {
8839   \xeCJK_if_ulem_patch:TF
8840   {
8841     \xeCJK_class_group_end:
8842     \UL@stop \xeCJK_ulem_hskip:n {#1} \UL@start
8843     \__xeCJK_ulem_class_group_begin:
8844     \xeCJK_select_punct_font:

```

```

8845     }
8846     { \__xeCJK_ulem_punct_breakable_kern:n {#1} }
8847 }

```

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

```

\__xeCJK_ulem_ccglue:
\__xeCJK_ulem_punct_ccglue:
8848 \cs_new_protected:Npn \__xeCJK_ulem_glue:n #1
8849 {
8850   \xeCJK_if_ulem_patch:TF
8851   {
8852     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
8853     { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
8854     {
8855       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
8856       { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
8857       { \skip_horizontal:n {#1} }
8858     }
8859   }
8860   { \skip_horizontal:n {#1} }
8861 }

```

改写 __xeCJK_use_ecglue_skip: (默认实现在 xeCJK 主体里是普通 \hskip), 让「源码空格 + 西文词」处补的 \CJKecglue 经 __xeCJK_ulem_glue:n 落到外层列表, 收缩量可参与断行(#1037)。

必须先用 \l__xeCJK_ulem_stream_started_bool 判断当前确实在装饰的 stream 里, 不能只依赖 __xeCJK_ulem_glue:n 自带的 \xeCJK_if_ulem_patch:TF。后者的判据只是 \ 的含义是否等于 ulem 保存的 \LA@space, 它能识别 ulem 自己造成的变化, 却无法区分「不在装饰中」与「在装饰外但 \ 被别的宏包改过定义」。

__xeCJK_check_for_ecglue_aux: 是所有中西文边界都会走的通用路径, 一旦在装饰外取到真分支, 就会在没有 \UL@box 打开的列表里执行 \UL@stop, 导致 Too many }'s。nath、morehype 等重定义 \ 的宏包会让不含任何装饰命令的中 abc 文直接报错。

该布尔由 __xeCJK_ulem_stream_begin: 置真、__xeCJK_ulem_end: 置假。

但它单独还不够: 行内公式里的装饰命令经 \UL@onmath/\UL@onin 结束, 不走 __xeCJK_ulem_end:, 因此同一个公式内装饰命令之后布尔仍为真, 而片段盒其实已经关闭。此时若再叠加 \ 被重定义, 仍会在没有 \UL@box 打开的列表里执行 \UL@stop——例如 $\$ \text{\CJKunderline{中} \text{\mbox{中 abc 文}}} \$$ 配 nath。因此还要确认片段盒确实打开: \UL@start 只在 ulem 打开片段盒后才被 \let 成 \@empty(ulem 的 \UL@start 定义末尾), 关闭后不再是它。 __xeCJK_ulem_exp_stop:w 已用同一判断表示这件事。

两个条件都成立才走 ulem 通道, 否则退化为主体的默认实现。

```

8862 \cs_gset_protected:Npn \__xeCJK_use_ecglue_skip:
8863 {
8864   \bool_lazy_and:nnTF
8865   { \l__xeCJK_ulem_stream_started_bool }
8866   { \cs_if_eq_p:NN \UL@start \@empty }
8867   { \__xeCJK_ulem_glue:n { \l__xeCJK_ecglue_skip } }
8868   { \skip_horizontal:N \l__xeCJK_ecglue_skip }
8869 }
8870 \cs_new_protected:Npn \xeCJK_make_group_tag:
8871 { \tl_set:Nc \l__xeCJK_group_tag_tl { \c__xeCJK_group_tag_tl } }
8872 \tl_new:N \l__xeCJK_group_tag_tl
8873 \tl_const:Nn \c__xeCJK_group_tag_tl
8874 {
8875   T \int_use:N \tex_currentgroupstype:D
8876   L \int_use:N \tex_currentgrouplevel:D
8877 }
8878 \cs_new_protected:Npn \__xeCJK_ulem_ccglue:
8879 { { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders } }
8880 \cs_new_protected:Npn \__xeCJK_ulem_punct_ccglue:
8881 { \__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` 会把正文拆进固定宽度盒子; `framework` 算出的左侧 `glue` 因此通过 `\_xeCJK_boundary_use_ulem_glue: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。

```

8882 \cs_new_protected:Npn \_xeCJK_boundary_use_ulem_glue:n #1
8883 {
8884   \xeCJK_if_ulem_patch:TF
8885   {
8886     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
8887     { \UL@stop \skip_horizontal:n {#1} \UL@start }
8888     {
8889       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
8890       { \UL@stop \skip_horizontal:n {#1} \UL@start }
8891       { \skip_horizontal:n {#1} }
8892     }
8893   }
8894   { \skip_horizontal:n {#1} }
8895 }
8896 \cs_new_protected:Npn \_xeCJK_ulem_stream_begin:
8897 {
8898   \bool_if:NF \l__xeCJK_ulem_stream_started_bool
8899   {
8900     \bool_set_true:N \l__xeCJK_ulem_stream_started_bool
8901     \_xeCJK_boundary_inline_stream_begin:nn { auto } { stream-ulem }
8902   }
8903 }
8904 \cs_new_protected:Npn \xeCJK_ulem_group_begin:
8905 {
8906   \mode_leave_vertical:
8907   \_xeCJK_ulem_stream_begin:
8908   \c_group_begin_token
8909 }
8910 \cs_new_protected:Npn \xeCJK_ulem_group_end:
8911 { \c_group_end_token }
8912 \cs_new_protected:Npn \xeCJK_ulem_on:n
8913 { \ULon }
8914 \cs_new_eq:NN \_xeCJK_orig_ulem_on: \ULon
8915 \cs_set_protected:Npn \ULon
8916 {
8917   \_xeCJK_ulem_stream_begin:
8918   \_xeCJK_orig_ulem_on:
8919 }
8920 \bool_new:N \l__xeCJK_ulem_stream_started_bool
8921 % 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
8922 % 装饰节点遮住。若正文语法确实是“公式+尾随源码空格”, 先让 ulem 排完
8923 % 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
8924 % 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,
8925 % 实际末节点检查仍会失败。
8926 %
8927 % 无论走哪条路径, 正文都必须以字面记号进入 \tn{UL@on} 的替换文本, 不能再经过
8928 % 一层宏参数间接展开: \pkg{ulem} 自己扫描正文并按源码空格切分片段, 正文若作为
8929 % 另一个宏的参数被重新扫描, 西文词右侧由边界恢复链补出的 \tn{CJKecglue} 会落在
8930 % 固定宽度的装饰片段盒内部, 其收缩量被盒子固化而无法参与断行, 整段可收缩量
8931 % 因此按西文词数减少 (\#1026)。
8932 %
8933 % 重排路径同样受这条约束。这里先把去掉尾随空格的正文和两端的固定记号拼成

```

```

8934 % 完整的 ulem 参数, 再用 \cs{exp_args:NV} 把它一次展开到参数位置, 使记号与
8935 % 直接书写 \texttt{\#1} 等价; 不能改用 \cs{tl_use:N}, 那会让正文晚一层展开而
8936 % 重新触发上述问题。
8937 % \tn{UL@on} 需要两端的 \cs{xeCJK_ulem_left:} / \cs{xeCJK_ulem_right:},
8938 % \tn{UL@onin} 由 \pkg{ulem} 自己的 \tn{UL@onin} 负责这两端, 因此拼装时把它们
8939 % 作为前后缀参数传入, 不写死在 body 里。
8940 \tl_new:N \l__xeCJK_ulem_body_tl
8941 \cs_new_protected:Npn \__xeCJK_boundary_ulem_math_tail_space:nnn #1#2#3
8942 {
8943   \tl_set:Nn \l__xeCJK_ulem_body_tl {#2}
8944   \tl_put_right:Ne \l__xeCJK_ulem_body_tl
8945     { \tl_trim_right_spaces:n {#1} }
8946   \tl_put_right:Nn \l__xeCJK_ulem_body_tl
8947     {
8948       \__xeCJK_boundary_if_last_math_node:T
8949       { \__xeCJK_boundary_math_end_confirm_space_visible: }
8950       \c_space_tl
8951     }
8952   \tl_put_right:Nn \l__xeCJK_ulem_body_tl {#3}
8953 }
8954 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_ulem_math_reorder:n #1
8955 { T , F , TF }
8956 {
8957   \__xeCJK_boundary_if_math_tail:nTF {#1}
8958   {
8959     \__xeCJK_boundary_if_math_tail_space:nTF {#1}
8960     { \prg_return_true: }
8961     { \prg_return_false: }
8962   }
8963   { \prg_return_false: }
8964 }
8965 \cs_new_eq:NN \__xeCJK_ulem_on:n \UL@on
8966 \cs_set_protected:Npn \UL@on #1
8967 {
8968   \__xeCJK_boundary_math_begin:n {#1}
8969   \__xeCJK_boundary_if_ulem_math_reorder:nTF {#1}
8970   {
8971     \__xeCJK_boundary_ulem_math_tail_space:nnn {#1}
8972     { \xeCJK_ulem_left: } { \xeCJK_ulem_right: }
8973     \exp_args:NV \__xeCJK_ulem_on:n \l__xeCJK_ulem_body_tl
8974   }
8975   {
8976     \__xeCJK_ulem_on:n
8977     {
8978       \xeCJK_ulem_left: #1
8979       \__xeCJK_boundary_math_end:n {#1}
8980       \xeCJK_ulem_right:
8981     }
8982   }
8983 }
8984 \cs_new_eq:NN \__xeCJK_orig_ulem_onin:n \UL@onin
8985 \cs_set_protected:Npn \UL@onin #1
8986 {
8987   \__xeCJK_boundary_math_begin:n {#1}
8988   \__xeCJK_boundary_if_ulem_math_reorder:nTF {#1}
8989   {
8990     \__xeCJK_boundary_ulem_math_tail_space:nnn {#1} { } { }
8991     \exp_args:NV \__xeCJK_orig_ulem_onin:n \l__xeCJK_ulem_body_tl
8992   }
8993   { \__xeCJK_orig_ulem_onin:n { #1 \__xeCJK_boundary_math_end:n {#1} } }
8994 }

```

\xeCJKfntefon 扩展 \ULon 的参数。

```

8995 \NewDocumentCommand \xeCJKfntefon { s t- s o }
8996 {
8997   \mode_leave_vertical:
8998   \xeCJK_ulem_boot:NNNn #1#2#3 {#4}

```

```

8999 \xeCJK_ulem_on:n
9000 }

\CJKunderline 9001 \NewDocumentCommand \CJKunderline { s t- s o }
9002 {
9003 \xeCJK_ulem_group_begin:
9004 \xeCJK_fntef_boot:nnNNNn { underline } { uline } #1#2#3 {#4}
9005 \cs_set_eq:NN \ULleaders \tex_cleaders:D
9006 \xeCJK_fntef_initial:nnn
9007 { \l__xeCJK_uline_depth_tl }
9008 { \l__xeCJK_uline_sep_tl }
9009 {
9010 \l__xeCJK_uline_format_tl
9011 \tex_vrule:D
9012 height \dim_eval:n { \l__xeCJK_uline_thickness_tl }
9013 depth \c_zero_dim
9014 width .2em
9015 }
9016 \xeCJK_ulem_on:n
9017 }
9018 \NewDocumentCommand \varCJKunderline { }
9019 { \CJKunderline - }

```

`\__xeCJK_fntef_wave_symbol:` 默认波浪符号来自固定的 lasy6 字号, 斜删除线来自当前数学字体; 两者的宽度与正文的字号和全角字宽没有稳定比例。ulem 又会在每个正文片段中重新排列装饰盒子, 因此固定宽度会让余量随字号、内容和伸缩间距改变。

这里用 l3draw 生成宽度为 1em/4 的完整波浪周期和斜线, 常规全角字符片段每字约容纳四个单元。两个默认符号都使用普通 \leaders。正文片段、可伸缩的 \CJKglue 和换行前后的片段虽然由 ulem 分开排出, 却仍以同一行的起点为相位基准, 因而不会在接点重新分配余量; 胶水仍保留原来的伸缩和断行能力。

普通形式保留 ulem 的既有几何, 即在正文左右各外伸半个图案周期。带减号形式则在正文左右各内缩半个周期; 两个相邻命令因此留下一个完整周期的断口, 同时不改变正文宽度。一般显式水平跳距仍按 ulem 的原有语义绘制装饰。绘图尺寸全部相对当前 em 计算, 因而会随字号缩放; 斜删除线覆盖基线以下约 .09em 到基线以上约 .84em, 与常见全角汉字的深度和高度相符, 也不再依赖数学字体。用户通过 underwave/symbol 指定的符号继续使用原来的 \xleaders; 任意自定义符号的周期、尺寸和接点不属于这里的默认图案处理范围。

```

9020 \cs_new_protected:Npn \__xeCJK_fntef_wave_symbol:
9021 {
9022 \dim_set:Nn \l__xeCJK_fntef_pattern_dim { 1 em / 4 }
9023 \hbox_to_wd:nn { \l__xeCJK_fntef_pattern_dim }
9024 {
9025 \hss
9026 \draw_begin:
9027 \draw_set_linewidth:n { .04 em }
9028 \draw_set_cap_but:
9029 \draw_path_moveto:n { 0 pt , .055 em }
9030 \draw_path_curveto:nnn
9031 { \l__xeCJK_fntef_pattern_dim * 3 / 16 , .055 em }
9032 { \l__xeCJK_fntef_pattern_dim * 5 / 16 , -.055 em }
9033 { \l__xeCJK_fntef_pattern_dim / 2 , -.055 em }
9034 \draw_path_curveto:nnn
9035 { \l__xeCJK_fntef_pattern_dim * 11 / 16 , -.055 em }
9036 { \l__xeCJK_fntef_pattern_dim * 13 / 16 , .055 em }
9037 { \l__xeCJK_fntef_pattern_dim , .055 em }
9038 \draw_path_use_clear:n { draw }
9039 \draw_end:
9040 \hss
9041 }
9042 }
9043 \cs_new_protected:Npn \__xeCJK_fntef_xout_symbol:
9044 {
9045 \dim_set:Nn \l__xeCJK_fntef_pattern_dim { 1 em / 4 }

```



```

9046 \hbox_to_wd:nn { \l__xeCJK_fntef_pattern_dim }
9047 {
9048   \hss
9049   \draw_begin:
9050     \draw_set_linewidth:n { .04 em }
9051     \draw_set_cap_but:
9052     \draw_path_moveto:n { 0 pt , -.445 em }
9053     \draw_path_lineto:n
9054       { \l__xeCJK_fntef_pattern_dim , .445 em }
9055     \draw_path_use_clear:n { draw }
9056   \draw_end:
9057   \hss
9058 }
9059 }
9060 \dim_new:N \l__xeCJK_fntef_pattern_dim

\CJKunderwave 9061 \NewDocumentCommand \CJKunderwave { s t- s o }
9062 {
9063   \xeCJK_ulem_group_begin:
9064     \xeCJK_fntef_boot:nnNNNn { underwave } { uwave } #1#2#3 {#4}
9065     \cs_set_eq:NN \ULleaders \tex_xleaders:D
9066     \tl_if_eq:NnT \l__xeCJK_uwave_symbol_tl { \__xeCJK_fntef_wave_symbol: }
9067     {
9068       \cs_set_eq:NN \ULleaders \tex_leaders:D
9069       \bool_set_true:N \l__xeCJK_ulem_periodic_bool
9070     }
9071     \xeCJK_fntef_initial:nnn
9072     { \l__xeCJK_uwave_depth_tl }
9073     { \l__xeCJK_uwave_sep_tl }
9074     { \l__xeCJK_uwave_format_tl \l__xeCJK_uwave_symbol_tl }
9075     \xeCJK_ulem_on:n
9076 }

\CJKunderdblline 9077 \NewDocumentCommand \CJKunderdblline { s t- s o }
9078 {
9079   \xeCJK_ulem_group_begin:
9080     \xeCJK_fntef_boot:nnNNNn { underdblline } { udblline } #1#2#3 {#4}
9081     \cs_set_eq:NN \ULleaders \tex_cleaders:D
9082     \xeCJK_fntef_initial:nnn
9083     { \l__xeCJK_udblline_depth_tl }
9084     { \l__xeCJK_udblline_sep_tl }
9085     {
9086       \l__xeCJK_udblline_format_tl
9087       \vbox_top:n
9088       {
9089         \tex_hrulerule:D
9090         height \dim_eval:n { \l__xeCJK_udblline_thickness_tl }
9091         depth \c_zero_dim
9092         width .2em
9093         \tex_kern:D \dim_eval:n { \l__xeCJK_udblline_gap_tl }
9094         \tex_hrulerule:D
9095         height \dim_eval:n { \l__xeCJK_udblline_thickness_tl }
9096         depth \c_zero_dim
9097         width .2em
9098       }
9099     }
9100     \xeCJK_ulem_on:n
9101 }

\JKsout 9102 \NewDocumentCommand \JKsout { s t- s o }
9103 {
9104   \xeCJK_ulem_group_begin:
9105     \xeCJK_fntef_boot:nnNNNn { sout } { sout } #1#2#3 {#4}
9106     \cs_set_eq:NN \ULleaders \tex_cleaders:D
9107     \xeCJK_fntef_initial:nn
9108     {
9109       \l__xeCJK_sout_format_tl
9110       \tex_vrule:D

```

```

9111         height \dim_eval:n { \l__xeCJK_sout_thickness_tl }
9112         depth \c_zero_dim
9113         width .2em
9114     }
9115     {
9116         \box_move_up:nn
9117         { \l__xeCJK_sout_height_tl - \box_ht:N \l__xeCJK_fntef_box / 2 }
9118         { \box_use:N \l__xeCJK_fntef_box }
9119     }
9120     \xeCJK_ulem_on:n
9121 }

```

```

\CJKxout 9122 \NewDocumentCommand \CJKxout { s t- s o }
9123 {
9124     \xeCJK_ulem_group_begin:
9125     \xeCJK_fntef_boot:nnNNn { xout } { xout } #1#2#3 {#4}
9126     \cs_set_eq:NN \ULleaders \tex_leaders:D
9127     \xeCJK_fntef_initial:nn
9128     {
9129         \l__xeCJK_xout_format_tl
9130         \__xeCJK_fntef_xout_symbol:
9131     }
9132     {
9133         \box_move_down:nn
9134         { .09 em }
9135         { \box_use:N \l__xeCJK_fntef_box }
9136     }
9137     \bool_set_true:N \l__xeCJK_ulem_periodic_bool
9138     \xeCJK_ulem_on:n
9139 }

```

```

\CJKunderanyline 9140 \NewDocumentCommand \CJKunderanyline { s t- s o m m }
9141 {
9142     \xeCJK_ulem_group_begin:
9143     \xeCJK_ulem_boot:NNNn #1#2#3 {#4}
9144     \cs_set_eq:NN \ULleaders \tex_cleaders:D
9145     \xeCJK_fntef_initial:nn
9146     {#6}
9147     {
9148         \box_move_down:nn
9149         {#5}
9150         { \box_use:N \l__xeCJK_fntef_box }
9151     }
9152     \tl_if_empty:NF \l__xeCJK_ulem_boxdepth_tl
9153     { \box_set_dp:Nn \ULC@box { \l__xeCJK_ulem_boxdepth_tl } }
9154     \tl_if_empty:NF \l__xeCJK_ulem_sep_tl
9155     {
9156         \bool_set_true:N \l__xeCJK_fntef_bool
9157         \dim_set:Nn \l__xeCJK_fntef_dim
9158         { \l__xeCJK_ulem_sep_tl + \box_dp:N \ULC@box }
9159     }
9160     \xeCJK_ulem_on:n
9161 }

```

\xeCJK_fntef_boot:nnNNn 处理参数问题。

```

9162 \cs_new_protected:Npn \xeCJK_fntef_boot:nnNNn #1#2#3#4#5#6
9163 {
9164     \bool_lazy_or:nnT {#3} {#5}
9165     { \bool_set_false:c { l__xeCJK_#2_skip_bool } }
9166     \bool_if:NT #4
9167     { \bool_set_true:c { l__xeCJK_#2_subtract_bool } }
9168     \tl_if_novalue:nF {#6}
9169     { \keys_set:nn { xeCJK / options / #1 } {#6} }
9170     \bool_set_eq:Nc \l__xeCJK_ulem_skip_bool { l__xeCJK_#2_skip_bool }
9171     \bool_set_eq:Nc \l__xeCJK_ulem_hidden_bool { l__xeCJK_#2_hidden_bool }
9172     \bool_set_eq:Nc \l__xeCJK_ulem_subtract_bool { l__xeCJK_#2_subtract_bool }
9173     \tl_set_eq:Nc \l__xeCJK_ulem_text_format_tl { l__xeCJK_#2_text_format_tl }

```

```

9174 }
9175 \cs_new_protected:Npn \xeCJK_ulem_boot:NNNn #1#2#3#4
9176 {
9177   \bool_lazy_or:nnT {#1} {#3}
9178   { \bool_set_false:N \l__xeCJK_ulem_skip_bool }
9179   \bool_if:NT #2
9180   { \bool_set_true:N \l__xeCJK_ulem_subtract_bool }
9181   \tl_if_novalue:nF {#4}
9182   { \keys_set:nn { xeCJK / options / ulem } {#4} }
9183 }

```

\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。

```

9184 \cs_new_protected:Npn \xeCJK_fntef_initial:n
9185 {
9186   \bool_if:NTF \l__xeCJK_nest_bool
9187   { \__xeCJK_warning:n { fntef-nesting } }
9188   {
9189     \bool_set_true:N \l__xeCJK_nest_bool
9190     \__xeCJK_restore_shipout_fntef:
9191   }
9192   \xeCJK_fntef_sbox:n
9193 }
9194 \cs_new_protected:Npn \xeCJK_fntef_initial:nn #1
9195 {
9196   \xeCJK_fntef_initial:n {#1}
9197   \bool_if:NF \l__xeCJK_fntef_bool
9198   { \dim_zero:N \l__xeCJK_fntef_dim }
9199   \markoverwith
9200 }
9201 \cs_new_protected:Npn \xeCJK_fntef_initial:nnn #1#2#3
9202 {
9203   \xeCJK_fntef_initial:n {#3}
9204   \bool_if:NF \l__xeCJK_fntef_bool
9205   {
9206     \bool_set_true:N \l__xeCJK_fntef_bool
9207     \dim_set:Nn \l__xeCJK_fntef_dim {#1}
9208   }
9209   \markoverwith
9210   {
9211     \box_move_down:nn
9212     { \l__xeCJK_fntef_dim + \box_ht:N \l__xeCJK_fntef_box }
9213     { \box_use:N \l__xeCJK_fntef_box }
9214   }
9215   \dim_set:Nn \l__xeCJK_fntef_dim { #2 + \box_dp:N \ULC@box }
9216 }
9217 \box_new:N \l__xeCJK_fntef_box
9218 \cs_new_eq:NN \xeCJKfntefbox \l__xeCJK_fntef_box
9219 \bool_new:N \l__xeCJK_nest_bool
9220 \bool_new:N \l__xeCJK_fntef_bool
9221 \__xeCJK_msg_new:nn { fntef-nesting }
9222 { 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。

```
9223 \cs_new_eq:NN \l__xeCJK_fntef_dim \ULdepth
```

\xeCJK_fntef_sbox:n 与 \hcoffin_set:Nn 和 L^AT_EX 2_ε 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。

装饰盒可能包含波浪线字符、数学斜线或者用户自定义的装饰符号,随后还会被 ulem 的 leaders 反复复制。用空的 ActualText 包住整个装饰盒,可以保留页面上的绘制结果,同时让 PDF 阅读器在复制和搜索时只看到正文。tagged PDF 会给盒子里的数学字符增加内层标记;若不暂停 tagging,内层标记仍会使斜删除线的字符进入文本提取结果。普通文档没有

\tag_suspend:n, 因此这里只在 LaTeX tagging 接口存在时暂停, 并在装饰内容结束后立即恢复。

```

9224 \cs_new_protected:Npn \xeCJK_fntef_sbox:n #1
9225 {
9226   \__xeCJK_boundary_capture_suspend:
9227   \hbox_set:Nn \l__xeCJK_fntef_box
9228   {
9229     \color_ensure_current:
9230     \BeginAccSupp { ActualText = { } }
9231     \cs_if_exist:NT \tag_suspend:n
9232     { \tag_suspend:n { xeCJKfntef } }
9233     #1
9234     \cs_if_exist:NT \tag_resume:n
9235     { \tag_resume:n { xeCJKfntef } }
9236     \EndAccSupp { }
9237   }
9238   \__xeCJK_boundary_capture_resume:
9239 }

```

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。

```

9240 \keys_define:nn { xeCJK / options }
9241 {
9242   underdot / symbol      .tl_set:N = \l__xeCJK_udot_symbol_tl ,
9243   underdot / depth      .tl_set:N = \l__xeCJK_udot_depth_tl ,
9244   underdot / sep        .tl_set:N = \l__xeCJK_udot_sep_tl ,
9245   underdot / format     .tl_set:N = \l__xeCJK_udot_format_tl ,
9246   underdot / textformat .tl_set:N = \l__xeCJK_udot_text_format_tl ,
9247   underdot / boxdepth   .tl_set:N = \l__xeCJK_udot_boxdepth_tl ,
9248   symbol / sep          .tl_set:N = \l__xeCJK_symbol_sep_tl ,
9249   symbol / boxdepth     .tl_set:N = \l__xeCJK_symbol_boxdepth_tl ,
9250   symbol / textformat   .tl_set:N = \l__xeCJK_symbol_text_format_tl ,
9251   underline / skip      .bool_set:N = \l__xeCJK_uline_skip_bool ,
9252   underline / hidden    .bool_set:N = \l__xeCJK_uline_hidden_bool ,
9253   underline / subtract  .bool_set:N = \l__xeCJK_uline_subtract_bool ,
9254   underline / thickness .tl_set:N = \l__xeCJK_uline_thickness_tl ,
9255   underline / depth     .tl_set:N = \l__xeCJK_uline_depth_tl ,
9256   underline / sep       .tl_set:N = \l__xeCJK_uline_sep_tl ,
9257   underline / format    .tl_set:N = \l__xeCJK_uline_format_tl ,
9258   underline / textformat .tl_set:N = \l__xeCJK_uline_text_format_tl ,
9259   underdblline / skip   .bool_set:N = \l__xeCJK_udbline_skip_bool ,
9260   underdblline / hidden .bool_set:N = \l__xeCJK_udbline_hidden_bool ,
9261   underdblline / subtract .bool_set:N = \l__xeCJK_udbline_subtract_bool ,
9262   underdblline / thickness .tl_set:N = \l__xeCJK_udbline_thickness_tl ,
9263   underdblline / depth  .tl_set:N = \l__xeCJK_udbline_depth_tl ,
9264   underdblline / sep    .tl_set:N = \l__xeCJK_udbline_sep_tl ,
9265   underdblline / format .tl_set:N = \l__xeCJK_udbline_format_tl ,
9266   underdblline / textformat .tl_set:N = \l__xeCJK_udbline_text_format_tl ,
9267   underdblline / gap    .tl_set:N = \l__xeCJK_udbline_gap_tl ,
9268   underwave / skip      .bool_set:N = \l__xeCJK_uwave_skip_bool ,
9269   underwave / hidden    .bool_set:N = \l__xeCJK_uwave_hidden_bool ,
9270   underwave / subtract  .bool_set:N = \l__xeCJK_uwave_subtract_bool ,
9271   underwave / symbol    .tl_set:N = \l__xeCJK_uwave_symbol_tl ,
9272   underwave / depth     .tl_set:N = \l__xeCJK_uwave_depth_tl ,
9273   underwave / sep       .tl_set:N = \l__xeCJK_uwave_sep_tl ,
9274   underwave / format    .tl_set:N = \l__xeCJK_uwave_format_tl ,
9275   underwave / textformat .tl_set:N = \l__xeCJK_uwave_text_format_tl ,
9276   sout / skip           .bool_set:N = \l__xeCJK_sout_skip_bool ,
9277   sout / hidden         .bool_set:N = \l__xeCJK_sout_hidden_bool ,
9278   sout / subtract       .bool_set:N = \l__xeCJK_sout_subtract_bool ,
9279   sout / thickness      .tl_set:N = \l__xeCJK_sout_thickness_tl ,
9280   sout / height         .tl_set:N = \l__xeCJK_sout_height_tl ,
9281   sout / format         .tl_set:N = \l__xeCJK_sout_format_tl ,
9282   sout / textformat     .tl_set:N = \l__xeCJK_sout_text_format_tl ,
9283   xout / skip           .bool_set:N = \l__xeCJK_xout_skip_bool ,

```

```

9284 xout / hidden .bool_set:N = \l__xeCJK_xout_hidden_bool ,
9285 xout / subtract .bool_set:N = \l__xeCJK_xout_subtract_bool ,
9286 xout / format .tl_set:N = \l__xeCJK_xout_format_tl ,
9287 xout / textformat .tl_set:N = \l__xeCJK_xout_text_format_tl ,
9288 ulem / skip .bool_set:N = \l__xeCJK_ulem_skip_bool ,
9289 ulem / hidden .bool_set:N = \l__xeCJK_ulem_hidden_bool ,
9290 ulem / subtract .bool_set:N = \l__xeCJK_ulem_subtract_bool ,
9291 ulem / sep .tl_set:N = \l__xeCJK_ulem_sep_tl ,
9292 ulem / boxdepth .tl_set:N = \l__xeCJK_ulem_boxdepth_tl ,
9293 ulem / textformat .tl_set:N = \l__xeCJK_ulem_text_format_tl
9294 }
9295 \clist_map_inline:nn
9296 { underdot , underline , underdblline , underwave , sout , xout , ulem }
9297 {
9298 \keys_define:nn { xeCJK / options }
9299 { #1 .meta:nn = { xeCJK / options / #1 } { ##1 } }
9300 }
9301 \keys_set:nn { xeCJK / options }
9302 {
9303 underdot / symbol = \normalfont . ,
9304 underdot / depth = 0.20 em ,
9305 underdot / sep = 0.04 em ,
9306 symbol / sep = \c_zero_dim ,
9307 underline / skip = true ,
9308 underline / thickness = \ULthickness ,
9309 underline / depth = 0.20 em ,
9310 underline / sep = 0.07 em ,
9311 underdblline / skip = true ,
9312 underdblline / thickness = \ULthickness ,
9313 underdblline / depth = 0.20 em ,
9314 underdblline / sep = 0.17 em ,
9315 underdblline / gap = 1.1 pt ,
9316 underwave / skip = true ,
9317 underwave / symbol = \__xeCJK_fntef_wave_symbol: ,
9318 underwave / depth = 0.20 em ,
9319 underwave / sep = 0.00 em ,
9320 sout / skip = true ,
9321 sout / thickness = \ULthickness ,
9322 sout / height = 0.35 em ,
9323 xout / skip = true
9324 }

```

```

\CJKunderanysymbol 9325 \NewDocumentCommand \CJKunderanysymbol { o m m m }
9326 {
9327 \xeCJK_under_symbol:nnnnnn { symbol } { symbol } {#1} {#2} {#3} {#4}
9328 }

```

\CJKunderdot \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。

```

9329 \NewDocumentCommand \CJKunderdot { o m }
9330 {
9331 \xeCJK_under_symbol:nnnnnn { underdot } { udot }
9332 {#1}
9333 { \l__xeCJK_udot_depth_tl }
9334 { \l__xeCJK_udot_format_tl \l__xeCJK_udot_symbol_tl }
9335 {#2}
9336 }

```

\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。

```

9337 \cs_new_protected:Npn \xeCJK_under_symbol:nnnnnn
9338 {
9339 \xeCJK_if_ulem_patch:TF
9340 { \__xeCJK_under_symbol_auxi:nnnnnn }
9341 { \__xeCJK_under_symbol_auxii:nnnnnn }
9342 }

```

```

9343 \cs_new_protected:Npn \__xeCJK_under_symbol_auxi:nnnnnn #1#2#3#4#5#6
9344 {
9345   \xeCJK_ulem_right: \UL@stop
9346   \group_begin:
9347     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
9348     \use:c { l__xeCJK_#2_text_format_tl }
9349     \UL@start \xeCJK_ulem_right_node:
9350     #6
9351     \xeCJK_ulem_right: \UL@stop
9352   \group_end:
9353   \UL@start \xeCJK_ulem_right_node:
9354 }
9355 \cs_new_protected:Npn \__xeCJK_under_symbol_auxii:nnnnnn #1#2#3#4#5#6
9356 {
9357   \__xeCJK_boundary_inline_stream_begin:nn { auto } { stream-ulem }
9358   \__xeCJK_boundary_math_begin:n {#6}
9359   \group_begin:
9360     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
9361     \__xeCJK_under_symbol_text_format:c { l__xeCJK_#2_text_format_tl }
9362     #6
9363     \__xeCJK_boundary_math_end:n {#6}
9364     \xeCJK_ulem_right:
9365   \group_end:
9366   \xeCJK_ulem_right_node:
9367   \__xeCJK_boundary_inline_stream_end:n { auto }
9368 }
9369 \cs_new_protected:Npn \xeCJK_under_symbol_initial:nnnnn #1#2#3#4#5
9370 {
9371   \tl_if_novalue:nF {#3}
9372   { \keys_set:nn { xeCJK / options / #1 } {#3} }
9373   \xeCJK_fntef_sbox:n {#5}
9374   \bool_if:NTF \l__xeCJK_fntef_bool
9375   { \xeCJK_make_under_symbol:n { \l__xeCJK_fntef_dim } }
9376   {
9377     \bool_set_true:N \l__xeCJK_fntef_bool
9378     \xeCJK_make_under_symbol:n {#4}
9379   }
9380   \tl_if_empty:cF { l__xeCJK_#2_boxdepth_tl }
9381   {
9382     \box_set_dp:Nn \l__xeCJK_under_symbol_box
9383     { \use:c { l__xeCJK_#2_boxdepth_tl } }
9384   }
9385   \dim_set:Nn \l__xeCJK_fntef_dim
9386   { \use:c { l__xeCJK_#2_sep_tl } + \box_dp:N \l__xeCJK_under_symbol_box }
9387   \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
9388   \__xeCJK_restore_shipout_CJKsymbol:
9389 }
9390 \cs_new_protected:Npn \__xeCJK_under_symbol_text_format:N #1
9391 {
9392   \tl_if_empty:NF #1
9393   { \xeCJK_ulem_right: #1 \xeCJK_ulem_right_node: }
9394 }
9395 \cs_generate_variant:Nn \__xeCJK_under_symbol_text_format:N { c }
9396 \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。

```

9397 \cs_new_protected:Npn \xeCJK_make_under_symbol:n #1
9398 {
9399   \hbox_set:Nn \l__xeCJK_under_symbol_box
9400   {
9401     \box_move_down:nn { #1 + \box_ht:N \l__xeCJK_fntef_box }
9402     {
9403       \hbox_to_zero:n
9404       {
9405         \xeCJK_select_font:
9406         \tex_kern:D \tex_fontcharwd:D \tex_font:D "4E00 \exp_stop_f:
9407         \tex_hss:D \box_use:N \l__xeCJK_fntef_box \tex_hss:D

```

```

9408     }
9409   }
9410 }
9411 }

```

`\_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。

```

9412 \cs_new_protected:Npn \_xeCJK_restore_shipout_CJKsymbol:
9413 {
9414   \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
9415     { \xeCJK_swap_cs:NN \CJKsymbol \_xeCJK_under_CJKsymbol:N }
9416   \_xeCJK_restore_shipout_fntef:
9417   \xeCJK_cs_clear:N \_xeCJK_restore_shipout_CJKsymbol:
9418 }
9419 \cs_new_protected:Npn \_xeCJK_restore_shipout_fntef:
9420 {
9421   \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
9422     {
9423       \bool_set_false:N \l__xeCJK_fntef_bool
9424       \dim_zero:N \l__xeCJK_fntef_dim
9425     }
9426   \xeCJK_cs_clear:N \_xeCJK_restore_shipout_fntef:
9427 }
9428 \tl_new:N \l__xeCJK_fntef_shipout_tl
9429 \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。

```

9430 \cs_new_protected:Npn \_xeCJK_under_CJKsymbol:N
9431 {
9432   \box_use:N \l__xeCJK_under_symbol_box
9433   \xeCJK_no_break: \_xeCJK_under_CJKsymbol:N
9434 }

```

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

```

9435 \NewDocumentEnvironment { CJKfilltwosides } { 0 { c } m }
9436 {
9437   \use:e { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
9438   \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
9439 }
9440 {
9441   \endminipage
9442   \ignorespacesafterend
9443 }
9444 \NewDocumentEnvironment { CJKfilltwosides* } { 0 { c } m +b }
9445 {
9446   \mode_leave_vertical:
9447   \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
9448   \tl_set:Nn \arraystretch { 1 }
9449   \cs_if_free:NF \extrarowheight
9450     { \cs_set_eq:NN \extrarowheight \c_zero_dim }
9451   \use:e { \_xeCJK_fill_two_sides:nnn {#1} { \dim_eval:n {#2} } } {#3}
9452 }
9453 { \ignorespacesafterend }
9454 \cs_new_protected:Npn \_xeCJK_fill_two_sides:nnn #1#2#3
9455 {
9456   \dim_compare:nNnTF {#2} > \c_zero_dim
9457   {
9458     \hbox_set:Nn \l__xeCJK_tmp_box
9459       { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
9460     \dim_compare:nNnTF {#2} > { \box_wd:N \l__xeCJK_tmp_box }
9461       { \tabular [#1] { @ { } p {#2} @ { } } #3 \endtabular }
9462       { \box_use:N \l__xeCJK_tmp_box }
9463   }

```

```

9464      { \tabular [#1] { @ { } c @ { } }          #3 \endtabular }
9465  }

```

\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。

```

9466 \cs_new_protected:Npn \xeCJK_fntef_hfilll:
9467   { \skip_horizontal:N \c__xeCJK_filll_skip }
9468 \skip_const:Nn \c__xeCJK_filll_skip { \c_zero_dim plus 1 filll }
9469 </fntef>

```

5.20 xeCJK-listings

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

```

9470 <*listings>
9471 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
9472 \ProcessOptions \scan_stop:
9473 \RequirePackage { xeCJK }
9474 \RequirePackage { listings }
9475 \lst@AddToHook { Init } { \__xeCJK_listings_initial_hook: }
9476 \lst@AddToHook { SelectCharTable } { \__xeCJK_listings_toks_hook: }
9477 \lst@AddToHook { OutputBox }
9478   {
9479     \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
9480     \l__xeCJK_restore_listings_toks_tl
9481     \__xeCJK_listings_output_CM:
9482   }
9483 \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。

```

9484 \cs_new_protected:Npn \__xeCJK_listings_initial_hook:
9485   {
9486     \tex_noindent:D
9487     \bool_gset_false:N \g__xeCJK_listings_CM_bool
9488     \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
9489     \xeCJK_add_to_shipout:n { \l__xeCJK_restore_listings_toks_tl }
9490     \lst@ifbreaklines
9491       \cs_set_eq:NN \__xeCJK_listings_CJK_toks_hook: \__xeCJK_listings_breaklines_toks:
9492       \tl_if_empty:NF \lst@prebreak
9493         { \tl_put_left:Nn \lst@prebreak { \l__xeCJK_restore_listings_toks_tl } }
9494       \tl_if_empty:NF \lst@postbreak
9495         { \tl_put_left:Nn \lst@postbreak { \l__xeCJK_restore_listings_toks_tl } }
9496     \fi:
9497     \int_set:Nn \l__xeCJK_listings_max_char_int
9498       { \lst@ifec 255 \else: 127 \fi: }
9499   }
9500 \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。

```

9501 \cs_new_protected:Npn \__xeCJK_listings_toks_hook:
9502   {
9503     \tl_clear:N \l__xeCJK_restore_listings_toks_tl
9504     \seq_map_function:NN
9505       \g__xeCJK_class_seq \__xeCJK_backup_inter_class_toks:n
9506       \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq

```



```

9507     {
9508         \str_if_eq:nnF { ##1 } { Boundary }
9509         {
9510             \xeCJK_inter_class_toks:nnn { Boundary } { ##1 }
9511             { \__xeCJK_listings_process_Default:nN { ##1 } }
9512         }
9513     }
9514     \xeCJK_inter_class_toks:nnn { Boundary } { CM }
9515     { \__xeCJK_listings_process_CM:nN { 0 } }
9516     \__xeCJK_listings_CJK_toks_hook:
9517 }

```

__xeCJK_backup_inter_class_toks:n 注意, 给 \XeTeXinterchartoks 赋空值, 会导致 XeTeX 崩溃!

```

9518 \cs_new_protected:Npn \__xeCJK_backup_inter_class_toks:n #1
9519 {
9520     \tl_set:Nx \l__xeCJK_tmp_tl
9521     { \xeCJK_get_inter_class_toks:nn { Boundary } {#1} }
9522     \tl_put_right:Nx \l__xeCJK_restore_listings_toks_tl
9523     {
9524         \xeCJK_inter_class_toks:nnn { Boundary } {#1}
9525         {
9526             \tl_if_empty:NTF \l__xeCJK_tmp_tl
9527             { \exp_not:N \prg_do_nothing: }
9528             { \exp_not:o \l__xeCJK_tmp_tl }
9529         }
9530     }
9531 }
9532 \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:

```

9533 \cs_new_protected:Npn \__xeCJK_listings_CJK_toks_hook:
9534 {
9535     \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
9536     { \__xeCJK_listings_process_CJK:nN { 2 } }
9537     \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
9538     { \__xeCJK_listings_process_CJK:nN { 2 } }
9539     \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
9540     { \__xeCJK_listings_process_CJK:nN { 2 } }
9541     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
9542     { \__xeCJK_listings_process_CJK:nN { 2 } }
9543     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
9544     { \__xeCJK_listings_process_CJK:nN { 2 } }
9545     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
9546     { \__xeCJK_listings_process_CM:nN { 0 } }
9547     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
9548     { \__xeCJK_listings_process_CM:nN { 0 } }
9549     \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
9550     { \__xeCJK_listings_process_CJK:nN { 2 } }
9551     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
9552     {
9553         \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
9554         { \__xeCJK_listings_process_CJK:nN { 2 } }
9555     }
9556 }
9557 \cs_new_protected:Npn \__xeCJK_listings_breaklines_toks:
9558 {
9559     \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
9560     { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9561     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
9562     { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9563     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
9564     { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9565     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
9566     { \__xeCJK_listings_process_CM:nN { 0 } }
9567     \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
9568     { \__xeCJK_listings_process_CM:nN { 0 } }

```

```

9569 \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
9570 { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9571 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
9572 { \_xeCJK_listings_process_FullLeft:nN { 2 } }
9573 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
9574 { \_xeCJK_listings_process_FullRight:nN { 2 } }
9575 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
9576 {
9577   \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
9578   { \_xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9579 }
9580 }

```

_xeCJK_listings_process_Default:nN 对于 \charcode 大于 255 的字符, 根据 \catcode 进行处理。

```

9581 \cs_new_protected:Npn \_xeCJK_listings_process_Default:nN #1#2
9582 {
9583   \int_compare:nNnTF
9584   { \xeCJK_token_value_charcode:N #2 } > \l__xeCJK_listings_max_char_int
9585   {
9586     \token_if_letter:NTF #2
9587     { \lst@ProcessLetter #2 }
9588     { \lst@ProcessOther #2 }
9589   }
9590   { \_xeCJK_listings_output_Default:nN {#1} #2 }
9591 }

```

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:。

```

9592 \cs_new_protected:Npn \_xeCJK_listings_output_Default:nN #1#2
9593 {
9594   \group_begin:
9595   \xeCJK_clear_inter_class_toks:nn { Boundary } {#1}
9596   \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \group_end: }
9597   #2
9598   \scan_stop:
9599 }

```

_xeCJK_listings_process_CJK:nN 对 CJK 字符类的处理。

```

9600 \cs_new_protected:Npn \_xeCJK_listings_process_CJK:nN #1#2
9601 {
9602   \token_if_letter:NTF #2
9603   { \_xeCJK_listings_process_letter:nN {#1} #2 }
9604   { \_xeCJK_listings_process_other:nN {#1} #2 }
9605 }

```

_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。

```

9606 \cs_new_protected:Npn \_xeCJK_listings_append:nN #1#2
9607 {
9608   \int_add:Nn \lst@length { #1 - 1 }
9609   \lst@Append #2
9610 }

```

_xeCJK_listings_process_letter:nN 在 letter 类中区分汉字和西文字母。

_xeCJK_listings_process_other:nN

```

9611 \cs_new_protected:Npn \_xeCJK_listings_process_letter:nN
9612 {
9613   \lst@whitespacefalse
9614   \bool_if:NTF \l__xeCJK_listings_letter_bool
9615   { \lst@lettertrue }
9616   {
9617     \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9618     \bool_set_true:N \l__xeCJK_listings_letter_bool

```

```

9619     }
9620     \__xeCJK_listings_append:nN
9621   }
9622   \bool_new:N \l__xeCJK_listings_letter_bool
9623   \cs_new_protected:Npn \__xeCJK_listings_process_other:nN #1#2
9624   {
9625     \lst@whitespacefalse
9626     \bool_if:NTF \l__xeCJK_listings_letter_bool
9627     {
9628       \lst@Output \lst@letterfalse
9629       \bool_set_false:N \l__xeCJK_listings_letter_bool
9630     }
9631     { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
9632     \cs_set_eq:NN \lst@lastother #2
9633     \__xeCJK_listings_append:nN {#1} #2
9634   }

```

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`。

```

9635   \cs_new_protected:Npn \__xeCJK_listings_process_breaklines_CJK:nN
9636   {
9637     \lst@whitespacefalse
9638     \bool_if:NTF \l__xeCJK_listings_letter_bool
9639     {
9640       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2 { \lst@Output }
9641       \lst@lettertrue
9642     }
9643     {
9644       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9645       \bool_set_true:N \l__xeCJK_listings_letter_bool
9646     }
9647     \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one_int
9648     \__xeCJK_listings_append:nN
9649   }
9650   \cs_new_protected:Npn \__xeCJK_listings_process_FullLeft:nN #1#2
9651   {
9652     \lst@whitespacefalse
9653     \bool_if:NTF \l__xeCJK_listings_letter_bool
9654     {
9655       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2
9656       {
9657         \int_compare:nNnTF \l__xeCJK_listings_flag_int = 3
9658         { \bool_if:NT \l__xeCJK_punct_breakable_bool { \lst@Output } }
9659         { \lst@Output }
9660       }
9661       \lst@lettertrue
9662     }
9663     {
9664       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9665       \bool_set_true:N \l__xeCJK_listings_letter_bool
9666     }
9667     \int_set:Nn \l__xeCJK_listings_flag_int { 2 }
9668     \__xeCJK_listings_append:nN {#1} #2
9669   }
9670   \cs_new_protected:Npn \__xeCJK_listings_process_FullRight:nN #1#2
9671   {
9672     \lst@whitespacefalse
9673     \bool_if:NTF \l__xeCJK_listings_letter_bool
9674     {
9675       \int_compare:nNnT \l__xeCJK_listings_flag_int < 2
9676       { \__xeCJK_punct_if_long:NT #2 { \lst@Output } }
9677       \lst@lettertrue
9678     }
9679     {
9680       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9681       \bool_set_true:N \l__xeCJK_listings_letter_bool

```

```

9682     }
9683     \int_set:Nn \l__xeCJK_listings_flag_int { 3 }
9684     \__xeCJK_listings_append:nN {#1} #2
9685   }
9686   \int_new:N \l__xeCJK_listings_flag_int

\lst@AppendLetter 9687 \cs_set_protected:Npn \lst@AppendLetter
\lst@AppendOther 9688 {
9689   \bool_if:NTF \l__xeCJK_listings_letter_bool
9690   {
9691     \lst@Output \lst@lettertrue
9692     \bool_set_false:N \l__xeCJK_listings_letter_bool
9693   }
9694   { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
9695   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
9696   \lst@Append
9697 }
9698 \cs_set_protected:Npn \lst@AppendOther
9699 {
9700   \bool_if:NTF \l__xeCJK_listings_letter_bool
9701   {
9702     \lst@Output \lst@letterfalse
9703     \bool_set_false:N \l__xeCJK_listings_letter_bool
9704   }
9705   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
9706   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
9707   \tex_futurelet:D \lst@lastother \lst@Append
9708 }

```

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

```

9709 \cs_new_protected:Npn \__xeCJK_listings_process_CM:nN
9710 {
9711   \reverse_if:N \lst@ifflexible
9712   \bool_gset_true:N \g__xeCJK_listings_CM_bool
9713   \fi:
9714   \__xeCJK_listings_process_letter:nN
9715 }

```

`\__xeCJK_listings_output_CM:` 在使用 `columns=fixed` 选项时, `listings` 会在输出盒子里的每个字符之间加入 `\hss`, 这就破坏了 $\text{\LaTeX}$ 将基本字和组合标识正确的组合起来。

```

9716 \cs_new_protected:Npn \__xeCJK_listings_output_CM:
9717 {
9718   \reverse_if:N \lst@ifflexible
9719   \bool_if:NT \g__xeCJK_listings_CM_bool
9720   {
9721     \bool_gset_false:N \g__xeCJK_listings_CM_bool
9722     \xeCJK_cs_clear:N \lst@FillOutputBox
9723     \cs_set_eq:NN \CJKglue \tex_hss:D
9724   }
9725   \fi:
9726 }
9727 \bool_new:N \g__xeCJK_listings_CM_bool

```

`\__xeCJK_listings_peek_active_loop:TF` `\lstinline` 通过判断参数中第一个字符是否是 active 类来区分它是否被用在其他宏的参数之中。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。

```

9728 \cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
9729 {
9730   \token_if_active:NTF #3
9731   { #1#3 }
9732   {
9733     \token_if_cs:NTF #3
9734     { #2#3 }
9735     {

```

```

9736         \int_compare:nNnTF {`#3} > { \l__xeCJK_listings_max_char_int }
9737         { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
9738         { #2#3 }
9739     }
9740 }
9741 }
9742 \cs_set_eq:NN \lst@ifnextcharactive \__xeCJK_listings_peek_active_loop:TF

```

__xeCJK_listings_rescan:Nn 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)。

```

9743 \cs_new_protected:Npn \__xeCJK_listings_rescan:Nn #1#2
9744 {
9745     \__xeCJK_listings_set_escape:
9746     \tl_set:Nn \l__xeCJK_tmp_tl {#2}
9747     \__xeCJK_listings_escape_backslash:
9748     \tl_clear:N \l__xeCJK_tmpb_tl
9749     \tl_map_inline:Nn \l__xeCJK_tmp_tl
9750     {
9751         \token_if_parameter:NTF ##1
9752         {
9753             \tl_put_right:Ne \l__xeCJK_tmpb_tl
9754             {
9755                 \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:N
9756                 \char_generate:nn { \int_value:w`##1 } { 13 }
9757             }
9758         }
9759         { \tl_put_right:Nn \l__xeCJK_tmpb_tl { ##1 } }
9760     }
9761     \tl_set_eq:NN \l__xeCJK_tmp_tl \l__xeCJK_tmpb_tl
9762     \tl_set_rescan:Nno #1
9763     { \char_set_catcode_active:n { 35 } }
9764     { \l__xeCJK_tmp_tl }
9765 }
9766 \cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
9767 {
9768     \__xeCJK_listings_rescan:Nn \l__xeCJK_tmp_tl {#1}
9769     \tl_put_right:No \lst@arg { \l__xeCJK_tmp_tl }
9770 }
9771 \cs_set_eq:NN \lst@insideconvert@ \__xeCJK_listings_inside_convert:nw
9772 \cs_new_protected:Npn \__xeCJK_listings_inline_group:w
9773 {
9774     \exp_after:wN \__xeCJK_listings_inline_group:n
9775     \exp_after:wN { \if_int_compare:w` = \c_zero_int \fi:
9776 }
9777 \cs_set_eq:NN \lst@inlineGJ \__xeCJK_listings_inline_group:w
9778 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
9779 {
9780     \__xeCJK_listings_rescan:Nn \lst@arg {#1}
9781     \lst@inlineGJEnd
9782 }

```

__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。

```

9783 \clist_map_inline:nn { \lstinline , \lstinline@ , \lst@InlineG , \lst@DeInit }
9784 { \__xeCJK_boundary_reserve:n {#1} }
9785 \bool_new:N \l__xeCJK_listings_boundary_bool
9786 \cs_new_protected:Npn \__xeCJK_listings_boundary_stream_end:
9787 { \__xeCJK_boundary_inline_stream_end:n { auto } }
9788 \cs_new_eq:NN \__xeCJK_orig_listings_inline:n \lstinline@
9789 \cs_set_protected:Npn \lstinline@ #1
9790 {
9791   \__xeCJK_boundary_inline_stream_begin:n { auto }
9792   \bool_set_true:N \l__xeCJK_listings_boundary_bool
9793   \__xeCJK_orig_listings_inline:n #1
9794 }
9795 \cs_new_eq:NN \__xeCJK_orig_listings_inline_group: \lst@InlineG
9796 \cs_set_protected:Npn \lst@InlineG
9797 {
9798   \__xeCJK_boundary_inline_stream_begin:n { auto }
9799   \bool_set_true:N \l__xeCJK_listings_boundary_bool
9800   \__xeCJK_orig_listings_inline_group:
9801 }
9802 \tl_gput_right:Nn \lst@DeInit
9803 {
9804   \bool_if:NT \l__xeCJK_listings_boundary_bool
9805     { \aftergroup \__xeCJK_listings_boundary_stream_end: }
9806 }

```

__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。

```

9807 \group_begin:
9808 \cs_set:Npn \__xeCJK_tmp:w #1
9809 {
9810   \group_end:
9811   \cs_new_protected:Npn \__xeCJK_listings_set_escape:
9812     { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
9813   \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
9814     { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
9815 }
9816 \use:n
9817 {
9818   \char_set_catcode_active:N \
9819   \__xeCJK_tmp:w
9820 }
9821 { \ }

```

__xeCJK_listings_escape_backslash: \catcode 为 12 的\ 需要双写转义。

```

9822 \cs_new_protected:Npe \__xeCJK_listings_escape_backslash:
9823 {
9824   \tl_replace_all:Nnn \exp_not:N \l__xeCJK_tmp_tl
9825     { \c_backslash_str }
9826     { \c_backslash_str \c_backslash_str }
9827 }
9828 </listings>
9829 <@@=xunadd>

```

5.21 xunicode-addon

```
9830 <*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。

```

9831 \bool_lazy_or:nnF
9832 { \sys_if_engine_xetex_p: }
9833 { \sys_if_engine luatex_p: }
9834 {
9835   \msg_new:nnnn { xunicode-addon } { xetex-luatex }
9836   { This~package~requires~either~XeTeX~or~LuaTeX~to~function.}
9837   {
9838     You~must~change~your~typesetting~engine~to,~e.g.,~\
9839     "xelatex"~or~"luaLatex"~instead~of~plain~"latex"~or~"pdflatex".
9840   }
9841   \msg_critical:n { xunicode-addon } { xetex-luatex }
9842 }
9843 \RequirePackage { xparse }

```

宏包选项是编码的名字。

```

9844 \clist_new:N \g__xunadd_encname_clist
9845 \tl_if_exist:NT \UTFencname
9846 { \clist_gput_right:Ne \g__xunadd_encname_clist { \UTFencname } }
9847 \DeclareOption*
9848 { \clist_gput_right:No \g__xunadd_encname_clist \CurrentOption }
9849 \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。

```

9850 \@ifpackageloaded { xunicode } { }
9851 {
9852   \clist_get:N NF \g__xunadd_encname_clist \UTFencname
9853   {
9854     \cs_if_exist:NTF \UnicodeEncodingName
9855     { \tl_set:N \UTFencname { \UnicodeEncodingName } }
9856     {
9857       \sys_if_engine_xetex:TF
9858       { \tl_set:N \UTFencname { EU1 } }
9859       { \tl_set:N \UTFencname { EU2 } }
9860     }
9861     \clist_gset_eq:NN \g__xunadd_encname_clist \UTFencname
9862   }
9863   \sys_if_engine_xetex:TF
9864   { \RequirePackage { xunicode } }
9865   {
9866     \cs_set_eq:NN \__xunadd_tmp:w \XeTeXpicfile
9867     \cs_set_eq:NN \XeTeXpicfile \prg_do_nothing:
9868     \RequirePackage { xunicode }
9869     \cs_set_eq:NN \XeTeXpicfile \__xunadd_tmp:w
9870   }
9871 }
9872 \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。

```

9873 \RenewDocumentCommand \ReloadXunicode { m }
9874 {
9875   \clist_set:N \l__xunadd_encname_clist {#1}
9876   \__xunadd_reload:N \l__xunadd_encname_clist
9877 }
9878 \cs_new_protected:Npn \__xunadd_reload:N #1
9879 {
9880   \cs_set_eq:NN \__xunadd_tmp:w \iftipaonetoken
9881   \cs_set_eq:NN \iftipaonetoken \scan_stop:
9882   \use:e
9883   {

```

```

9884 \ExplSyntaxOff
9885 \char_set_catcode_letter:n { 64 }
9886 \exp_not:N \clist_map_function:NN \exp_not:N #1 \__xunadd_reload_aux:n
9887 \bool_if:NTF \l__kernel_expl_bool
9888 { \ExplSyntaxOn }
9889 { \ExplSyntaxOff }
9890 \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
9891 }
9892 \cs_set_eq:NN \iftipaonetoken \__xunadd_tmp:w
9893 }
9894 \cs_new_protected:Npn \__xunadd_reload_aux:n #1
9895 {
9896 \cs_if_exist:cTF { T@ #1 }
9897 {
9898 \tl_set:Nn \UTFencname {#1}
9899 \clist_if_in:NnF \g__xunadd_encname_clist {#1}
9900 { \clist_gput_right:Nn \g__xunadd_encname_clist {#1} }
9901 \file_input:n { xunicode.sty }
9902 \file_input:n { xunicode-extra.def }
9903 }
9904 { \msg_error:nnn { xunicode-addon } { encoding-unknown } {#1} }
9905 }
9906 \clist_new:N \l__xunadd_encname_clist
9907 \msg_new:nnnn { xunicode-addon } { encoding-unknown }
9908 { Encoding~scheme~"#1"~unknown. }
9909 {
9910 You~may~use \\\
9911 \token_to_str:N \usepackage [ #1 , \encodingdefault ] \{fontenc\} \\\
9912 before~xunicode-addon~or~xunicode.
9913 }

```

\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。

```

9914 \RenewDocumentCommand \DeclareUTFmathsymbols { m }
9915 {
9916 \bool_if:NT \l__xunadd_math_as_UTF_text_bool
9917 {
9918 \seq_map_inline:Nn \l__xunadd_math_as_UTF_text_seq
9919 { \__xunadd_declare_math_as_UTF_text:n {##1} }
9920 \bool_set_false:N \l__xunadd_math_as_UTF_text_bool
9921 }
9922 }
9923 \seq_new:N \l__xunadd_math_as_UTF_text_seq
9924 \seq_set_from_clist:Nn \l__xunadd_math_as_UTF_text_seq
9925 { hbar , Finv , aleph , beth , gimel , daleth , Game }
9926 \bool_new:N \l__xunadd_math_as_UTF_text_bool
9927 \RenewDocumentCommand \UseMathAsText { }
9928 {
9929 \math@s@text@true
9930 \bool_set_true:N \l__xunadd_math_as_UTF_text_bool
9931 }
9932 \@onlypreamble \UseMathAsText
9933 \cs_new_protected:Npn \__xunadd_declare_math_as_UTF_text:n #1
9934 {
9935 \cs_if_exist:cTF {#1}
9936 {
9937 \cs_new_eq:cc { keepmathUTF #1 } {#1}
9938 \cs_gset_protected:cpe {#1}
9939 {
9940 \exp_not:N \mode_if_math:TF
9941 { \exp_not:c { keepmathUTF #1 } }
9942 { \exp_not:c { text #1 } }
9943 }
9944 \tl_put_right:Ne \l__xunadd_hyperref_hook_tl
9945 { \cs_set_eq:NN \exp_not:c {#1} \exp_not:c { text #1 } }
9946 }
9947 { \cs_new:cpe {#1} { \exp_not:c { text #1 } } }
9948 }

```



```

9949 \tl_new:N \l__xunadd_hyperref_hook_tl
9950 \AtBeginDocument
9951 {
9952   \cs_if_free:NF \pdfstringdefDisableCommands
9953   { \pdfstringdefDisableCommands { \l__xunadd_hyperref_hook_tl } }
9954 }

```

判断字符在当前字体中是否存在。

```

9955 \prg_new_conditional:Npnn \__xunadd_glyph_if_exist:n #1 { p , T , F , TF }
9956 {
9957   \tex_iffontchar:D \tex_font:D \tex_numexpr:D #1 \scan_stop:
9958   \prg_return_true: \else: \prg_return_false: \fi:
9959 }

```

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

```

9960 \RenewDocumentCommand \UndeclareUTFcharacter { O { \UTFencname } m m }
9961 {
9962   \__xunadd_if_csname:nTF {#3}
9963   { \UndeclareTextCommand {#3} }
9964   { \exp_args:Nc \UndeclareTextCommand { \tl_to_str:n {#3} } }
9965   {#1}
9966 }

```

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

```

9967 \RenewDocumentCommand \UndeclareUTFcomposite { O { \UTFencname } m m m }
9968 {
9969   \__xunadd_if_csname:nTF {#3}
9970   { \__xunadd_undeclare_composite:Nnnn #3 }
9971   { \exp_args:Nc \__xunadd_undeclare_composite:Nnnn { \tl_to_str:n {#3} } }
9972   {#1} {#4} {#2}
9973 }
9974 \cs_new_protected:Npn \__xunadd_undeclare_composite:Nnnn #1#2#3#4
9975 { \cs_undefine:c { \__xunadd_composite_cs:Nnn #1 {#2} {#3} } }

```

```

9976 \cs_new:Npe \__xunadd_composite_cs:Nnn #1#2#3
9977 { \c_backslash_str #2 \exp_not:N \token_to_str:N #1 - \exp_not:N \tl_to_str:n {#3} }
9978 \cs_new:Npe \__xunadd_composite_cs:nnn #1#2#3
9979 { \c_backslash_str #2 #1 - \exp_not:N \tl_to_str:n {#3} }

```

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}

9980 \prg_new_conditional:Npnn \__xunadd_if_csname:n #1 { TF }
9981 {
9982   \tl_if_single_token:nTF {#1}
9983   {
9984     \token_if_cs:NTF #1
9985     { \prg_return_true: }
9986     {
9987       \token_if_active:NTF #1
9988       { \prg_return_true: }
9989       { \prg_return_false: }
9990     }
9991   }
9992   { \prg_return_false: }
9993 }

```

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

```

9994 \RenewDocumentCommand \DeclareUTFcharacter { O { \UTFencname } m m }
9995 {
9996   \cs_if_exist_use:cF
9997   { __xunadd_restore_ \tl_to_str:n {#3} : }
9998   {
9999     \__xunadd_if_csname:nTF {#3}
10000     { \__xunadd_declare_character:Nnn #3 }

```

```

10001      { \__xunadd_declare_character:cnn { \tl_to_str:n {#3} } }
10002      {#1} {#2}
10003    }
10004  }

```

__xunadd_restore_cmd:N 恢复 \hbar 和 \nobreakspace 为原本定义。

```

10005 \cs_new_protected:cpn
10006 { \__xunadd_restore_ \tl_to_str:n { \hbar } : }
10007 { \__xunadd_restore_cmd:N \hbar }
10008 \cs_new_protected:cpn
10009 { \__xunadd_restore_ \tl_to_str:n { \nobreakspace } : }
10010 { \__xunadd_restore_cmd:N \nobreakspace }
10011 \cs_new_protected:Npn \__xunadd_restore_cmd:N #1
10012 { \__xunadd_restore_cmd:Ne #1 { ? - \token_to_str:N #1 } }
10013 \cs_new_protected:Npn \__xunadd_restore_cmd:Nn #1#2
10014 {
10015   \cs_if_free:cF {#2}
10016   { \__xunadd_restore_cmd:Nc #1 {#2} }
10017 }
10018 \cs_new_protected:Npn \__xunadd_restore_cmd:NN #1#2
10019 {
10020   \cs_gset_eq:NN #1 #2
10021   \cs_undefine:N #2
10022 }
10023 \cs_generate_variant:Nn \__xunadd_restore_cmd:Nn { Ne }
10024 \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。

```

10025 \cs_new_protected:Npn \__xunadd_declare_character:Nnn #1#2#3
10026 {
10027   \__xunadd_provide_text_command_default:N #1
10028   \exp_after:wN \__xunadd_declare_character:Nnen
10029   \tex_Uchar:D \__xunadd_check_slot:n {#3} \exp_stop_f:
10030   #1 { \token_to_str:N #1 } {#2}
10031 }
10032 \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
10033 \NewDocumentCommand \DeclareUTFSymbol { m O { \UTFencname } m }
10034 { \__xunadd_declare_character:Nnn #1 {#2} {#3} }
10035 \NewDocumentCommand \DeclareUTFCommand { m O { \UTFencname } m }
10036 { \__xunadd_text_command:Nnn #1 { \token_to_str:N #1 } {#2} {#3} }
10037 \cs_new_protected:Npn \__xunadd_text_command:Nnnn #1#2#3#4
10038 { \DeclareTextCommand #1 {#3} { \__xunadd_text_command:nn {#2} {#4} } }
10039 \cs_generate_variant:Nn \__xunadd_text_command:Nnnn { No }
10040 \cs_new_protected:Npn \__xunadd_text_command:nn #1#2
10041 {
10042   \__xunadd_begin_hook:nn {#1} {#2}
10043   #2
10044   \__xunadd_end_hook:nn {#1} {#2}
10045 }

```

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

```

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

```

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

```

10046 \cs_new_protected:Npn \__xunadd_provide_text_command_default:N #1
10047 {
10048   \cs_if_exist:cF { ? \token_to_str:N #1 }
10049   {
10050     \cs_if_free:cF { ? - \token_to_str:N #1 }
10051     {
10052       \exp_args:NNv \ProvideTextCommandDefault #1
10053       { ? - \token_to_str:N #1 }
10054     }
10055   }
10056 }

```

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。

```

10057 \cs_new_protected:Npn \__xunadd_declare_character:NNnn #1#2#3#4
10058 { \DeclareTextCommand #2 {#4} { \__xunadd_text_character:nN {#3} {#1} } }
10059 \cs_new_protected:Npn \__xunadd_text_character:nN #1#2
10060 {
10061   \__xunadd_begin_hook:nn {#1} {#2}
10062   \__xunadd_glyph_if_exist:nTF { `#2 }
10063   {#2} { \cs_if_exist_use:cF { ? #1 } {#2} }
10064   \__xunadd_end_hook:nn {#1} {#2}
10065 }
10066 \cs_generate_variant:Nn \__xunadd_declare_character:NNnn { NNe }

```

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

```

10067 \cs_new:Npn \__xunadd_check_slot:n #1
10068 {
10069   \int_eval:n
10070   {
10071     \tl_if_head_eq_charcode:nNTF {#1} x
10072     { " \use_none:n #1 } {#1}
10073   }
10074 }

```

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。

```

10075 \RenewDocumentCommand \DeclareUTFcomposite { 0 { \UTFencname } m m m }
10076 {
10077   \__xunadd_if_csname:nTF {#3}
10078   { \__xunadd_declare_composite:Nnnn #3 }
10079   { \__xunadd_declare_composite:cnnn { \tl_to_str:n {#3} } }
10080   {#1} {#4} {#2}
10081 }

```

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。

```

10082 \cs_new_protected:Npn \__xunadd_declare_composite:Nnnn #1#2#3#4
10083 {
10084   \tex_afterassignment:D \use_none_delimit_by_q_stop:w
10085   \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
10086   { \__xunadd_check_slot:n {#4} }
10087   \q_stop
10088 }
10089 \cs_new_protected:Npn \__xunadd_chardef:Nn #1#2
10090 { \tex_chardef:D #1 = \tex_numexpr:D #2 \scan_stop: }
10091 \cs_generate_variant:Nn \__xunadd_chardef:Nn { c }
10092 \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。
10093	\NewDocumentCommand \DeclareUTFCompositeCommand { m O { \UTFencname } m m }
10094	{ \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。
10095	\NewDocumentCommand \DeclareUTFCompositeSymbol { m O { \UTFencname } m m }
10096	{
10097	__xunadd_chardef:cn { __xunadd_composite_cs:Nnn #1 {#2} {#3} }
10098	{ __xunadd_check_slot:n {#4} }
10099	}
\DeclareUTFComposite	将 #1 设置为编码 #2 下的带一个参数的复合符号命令。
10100	\NewDocumentCommand \DeclareUTFComposite { m O { \UTFencname } }
10101	{ \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。
10102	\NewDocumentCommand \DeclareUTFEncodedAccent { m O { \UTFencname } m m }
10103	{ __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。
10104	\NewDocumentCommand \DeclareUTFEncodedAccents { m O { \UTFencname } m m }
10105	{ __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。
10106	\NewDocumentCommand \DeclareUTFEncodedSymbol { m O { \UTFencname } m m }
10107	{ __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。
10108	\NewDocumentCommand \DeclareUTFEncodedCircle { m O { \UTFencname } m m }
10109	{ __xunadd_declare_encoded:NNnnn __xunadd_combine_circle:nnNNn #1 {#2} {#3} {#4} }
\DeclareEncodedCompositeCharacter	10110 \RenewDocumentCommand \DeclareEncodedCompositeCharacter { m m m m }
10111	{ \DeclareUTFEncodedSymbol #2 [#1] { "#3 } { "0#4 } }
\DeclareEncodedCompositeAccents	10112 \RenewDocumentCommand \DeclareEncodedCompositeAccents { m m m m }
10113	{ \DeclareUTFEncodedAccents #2 [#1] { "#4 } { "#3 } }
\DeclareUTFDoubleEncodedAccent	10114 \NewDocumentCommand \DeclareUTFDoubleEncodedAccent { m O { \UTFencname } m m }
10115	{ __xunadd_declare_encoded:NNnnn __xunadd_combine_double_accent:nnNNn #1 {#2} {#3} {#4} }
\DeclareUTFDoubleEncodedSymbol	10116 \NewDocumentCommand \DeclareUTFDoubleEncodedSymbol { m O { \UTFencname } m m }
10117	{ __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。
10118	\cs_new_protected:Npn __xunadd_declare_composite:Nnn #1#2#3
10119	{ \DeclareTextCommand #1 {#3} { __xunadd_text_composite:nnn {#2} {#3} } }
__xunadd_text_composite:nnn	10120 \cs_new_protected:Npn __xunadd_text_composite:nnn #1#2#3
10121	{
10122	__xunadd_begin_hook:nn {#1} {#3}
10123	\cs_if_exist:cTF { __xunadd_composite_cs:nnn {#1} {#2} {#3} }
10124	{
10125	__xunadd_text_composite:cnn
10126	{ __xunadd_composite_cs:nnn {#1} {#2} {#3} } {#1} {#3}
10127	}
10128	{ \cs_if_exist_use:cTF { ? #1 } { {#3} } {#3} }
10129	__xunadd_end_hook:nn {#1} {#3}

```

10130 }
10131 \cs_new_protected:Npn \__xunadd_text_composite:Nnn #1#2#3
10132 {
10133   \token_if_chardef:NTF #1
10134   {
10135     \__xunadd_glyph_if_exist:nTF {#1}
10136     {#1} { \cs_if_exist_use:cTF { ? #2 } { {#3} } {#3} }
10137   }
10138   {#1}
10139 }
10140 \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。

```

10141 \cs_new_protected:Npn \__xunadd_declare_encoded:NNnnn #1#2#3#4#5
10142 {
10143   \exp_after:wN \__xunadd_declare_encoded:NNNNee
10144   \tex_Uchar:D \__xunadd_check_slot:n {#4} \exp_after:wN \exp_stop_f:
10145   \tex_Uchar:D \__xunadd_check_slot:n {#5} \exp_stop_f:
10146   #1 #2 { \token_to_str:N #2 } {#3}
10147 }
10148 \cs_new_protected:Npn \__xunadd_declare_encoded:NNNNnn #1#2#3#4#5#6
10149 { \DeclareTextCommand #4 {#6} { #3 {#5} {#6} {#1} {#2} } }
10150 \cs_generate_variant:Nn \__xunadd_declare_encoded:NNnnn { c }
10151 \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。

```

10152 \cs_new_protected:Npn \__xunadd_text_combine:NnnNNn #1#2#3#4#5#6
10153 {
10154   \__xunadd_begin_hook:nn {#2} {#6}
10155   \cs_if_exist:cTF { \__xunadd_composite_cs:nnn {#2} {#3} {#6} }
10156   {
10157     \__xunadd_text_combine:cNnNNn
10158     { \__xunadd_composite_cs:nnn {#2} {#3} {#6} } #1 {#2} {#4} {#5} {#6}
10159   }
10160   { #1 {#6} {#2} {#4} {#5} }
10161   \__xunadd_end_hook:nn {#2} {#6}
10162 }
10163 \cs_new_protected:Npn \__xunadd_text_combine:NNnNNn #1#2#3#4#5#6
10164 {
10165   \token_if_chardef:NTF #1
10166   { \__xunadd_glyph_if_exist:nTF {#1} {#1} { #2 {#6} {#3} {#4} {#5} } }
10167   {#1}
10168 }
10169 \cs_generate_variant:Nn \__xunadd_text_combine:NNnNNn { c }

```

```

\__xunadd_combine_symbol:nnNNn 10170 \cs_new_protected:Npn \__xunadd_combine_symbol:nnNNn
10171 { \__xunadd_text_combine:NnnNNn \__xunadd_add_symbol:nnNN }
10172 \cs_new_protected:Npn \__xunadd_add_symbol:nnNN #1#2#3#4
10173 {
10174   \tl_if_blank:nTF {#1}
10175   {
10176     \__xunadd_glyph_if_exist:nTF { `#4 }
10177     {#4}
10178     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10179   }
10180   {
10181     \__xunadd_glyph_if_exist:nTF { `#3 }
10182     { #1#3 }
10183     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10184   }
10185 }

```

__xunadd_combine_accent:nnNNn 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。
 __xunadd_add_accent:nnNN

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。

```

10186 \cs_new_protected:Npn \__xunadd_combine_accent:nnNNn
10187 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accent:nnNN }
10188 \cs_new_protected:Npn \__xunadd_add_accent:nnNN #1#2#3#4
10189 {
10190   \tl_if_blank:nTF {#1}
10191   {
10192     \__xunadd_glyph_if_exist:nTF {`#4 }
10193     {#4}
10194     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10195   }
10196   {
10197     \__xunadd_glyph_if_exist:nTF {`#3 }
10198     { #1#3 }
10199     {
10200       \__xunadd_glyph_if_exist:nTF {`#4 }
10201       { \add@accent {`#4 } {#1} }
10202       { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10203     }
10204   }
10205 }

```

```

\__xunadd_combine_accents:nnNNn 10206 \cs_new_protected:Npn \__xunadd_combine_accents:nnNNn
\__xunadd_add_accents:nnNN 10207 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accents:nnNN }
10208 \cs_new_protected:Npn \__xunadd_add_accents:nnNN #1#2#3#4
10209 {
10210   \tl_if_blank:nTF {#1}
10211   { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
10212   {
10213     \__xunadd_glyph_if_exist:nTF {`#3 }
10214     { \__xunadd_glyph_if_exist:nTF {`#4 } }
10215     { \use_ii:nn }
10216     { #1#3#4 }
10217     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3#4 } }
10218   }
10219 }

```

`\__xunadd_combine_circle:nnNNn` 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。

```

\__xunadd_add_circle:nnNNn
\__xunadd_add_circle:nN
10220 \cs_new_protected:Npn \__xunadd_combine_circle:nnNNn
10221 { \__xunadd_text_combine:NnnNNn \__xunadd_add_circle:nnNN }
10222 \cs_new_protected:Npn \__xunadd_add_circle:nnNN #1#2#3#4
10223 {
10224   \tl_if_blank:nTF {#1}
10225   {
10226     \__xunadd_glyph_if_exist:nTF {`#4 }
10227     {#4}
10228     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10229   }
10230   {
10231     \__xunadd_glyph_if_exist:nTF {`#4 }
10232     { \__xunadd_add_circle:nN {#1} #4 }
10233     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
10234   }
10235 }
10236 \cs_new_protected:Npn \__xunadd_add_circle:nN #1#2
10237 {
10238   \hcoffin_set:Nn \l__xunadd_tmp_coffin {#1}
10239   \hcoffin_set:Nn \l__xunadd_circle_coffin {#2}
10240   \fp_set:Nn \l__xunadd_circle_scale_fp

```

¹⁹<http://tug.org/pipermail/xetex/2013-July/024579.html>

```

10241     {
10242         \dim_to_decimal_in_unit:nn
10243         {
10244             \fp_use:N \l__xunadd_circle_ratio_fp
10245             \coffin_wd:N \l__xunadd_circle_coffin
10246         }
10247         { \coffin_wd:N \l__xunadd_tmp_coffin }
10248     }
10249     \coffin_scale:Nnn \l__xunadd_tmp_coffin
10250     { \l__xunadd_circle_scale_fp } { \l__xunadd_circle_scale_fp }
10251     \coffin_attach:NnnNnnnn
10252     \l__xunadd_circle_coffin { hc } { vc }
10253     \l__xunadd_tmp_coffin { hc } { vc } { \c_zero_dim } { \c_zero_dim }
10254     \coffin_typeset:Nnnnn \l__xunadd_circle_coffin
10255     { H } { l } { \c_zero_dim } { \c_zero_dim }
10256 }
10257 \fp_new:N \l__xunadd_circle_scale_fp
10258 \coffin_new:N \l__xunadd_tmp_coffin
10259 \coffin_new:N \l__xunadd_circle_coffin

```

`\settextcircledratio` 设置圆圈中文字的宽度与圆圈宽度的比例, 预设为 0.7。

```

10260 \NewDocumentCommand \settextcircledratio { m }
10261 { \fp_set:Nn \l__xunadd_circle_ratio_fp {#1} }
10262 \fp_new:N \l__xunadd_circle_ratio_fp
10263 \settextcircledratio { 0.7 }

```

`\__xunadd_combine_double_accent:nnNNn` 使 `\t` 等组合重音符号放在参数的第一个字母的右边。

```

10264 \cs_new_protected:Npn \__xunadd_combine_double_accent:nnNNn
10265 { \__xunadd_text_combine:NnnNNn \__xunadd_add_double_accent:nnNN }
10266 \cs_new_protected:Npn \__xunadd_add_double_accent:nnNN #1#2#3#4
10267 {
10268     \tl_if_blank:nTF {#1}
10269     {
10270         \__xunadd_glyph_if_exist:nTF { `#4 }
10271         {#4}
10272         { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10273     }
10274     {
10275         \__xunadd_glyph_if_exist:nTF { `#3 }
10276         { \__xunadd_add_double_symbol:nN {#1} #3 }
10277         {
10278             \__xunadd_glyph_if_exist:nTF { `#4 }
10279             { \add@accent { `#4 } {#1} }
10280             { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10281         }
10282     }
10283 }

```

`\__xunadd_combine_double_symbol:nnNNn` 使 `\sliding` 等组合重音符号放在参数的第一个字母的右边。

```

10284 \cs_new_protected:Npn \__xunadd_combine_double_symbol:nnNNn
10285 { \__xunadd_text_combine:NnnNNn \__xunadd_add_double_symbol:nnNN }
10286 \cs_new_protected:Npn \__xunadd_add_double_symbol:nnNN #1#2#3#4
10287 {
10288     \tl_if_blank:nTF {#1}
10289     {
10290         \__xunadd_glyph_if_exist:nTF { `#4 }
10291         {#4}
10292         { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10293     }
10294     {
10295         \__xunadd_glyph_if_exist:nTF { `#3 }
10296         { \__xunadd_add_double_symbol:nN {#1} #3 }
10297         { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10298     }
10299 }

```

`\_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。

```

10300 \cs_new_protected:Npn \_xunadd_add_double_symbol:nN #1#2
10301 {
10302   \tl_if_head_is_N_type:nTF {#1}
10303   {
10304     \exp_after:wN \exp_after:wN \exp_after:wN
10305     \_xunadd_add_double_symbol_aux:NnN \exp_after:wN \exp_after:wN
10306     \tl_head:w #1 \q_stop \exp_after:wN { \use_none:n #1 } #2
10307   }
10308   { #1#2 }
10309 }
10310 \cs_new_protected:Npn \_xunadd_add_double_symbol_aux:NnN #1#2#3
10311 {
10312   \bool_lazy_any:nTF
10313   {
10314     { \token_if_letter_p:N #1 }
10315     { \token_if_other_p:N #1 }
10316     { \token_if_chardef_p:N #1 }
10317   }
10318   { #1#3#2 }
10319   { #1#2#3 }
10320 }

```

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

```

10321 \NewDocumentCommand \AtBeginUTFCommand { s O { } +m }
10322 {
10323   \tl_if_blank:nTF {#2}
10324   {
10325     \IfBooleanTF {#1}
10326     { \xunadd_set_begin_hook:n }
10327     { \xunadd_append_begin_hook:n }
10328   }
10329   { \xunadd_set_begin_hook:nn {#2} }
10330   {#3}
10331 }
10332 \NewDocumentCommand \AtEndUTFCommand { s O { } +m }
10333 {
10334   \tl_if_blank:nTF {#2}
10335   {
10336     \IfBooleanTF {#1}
10337     { \xunadd_set_end_hook:n }
10338     { \xunadd_append_end_hook:n }
10339   }
10340   { \xunadd_set_end_hook:nn {#2} }
10341   {#3}
10342 }

```

```

\xunadd_set_begin_hook:n 10343 \cs_new_protected:Npn \xunadd_set_begin_hook:n
\xunadd_set_end_hook:n 10344 { \tl_set:Nn \l__xunadd_begin_hook_tl }
10345 \cs_new_protected:Npn \xunadd_append_begin_hook:n
10346 { \tl_put_right:Nn \l__xunadd_begin_hook_tl }
10347 \cs_new_protected:Npn \xunadd_set_end_hook:n
10348 { \tl_set:Nn \l__xunadd_end_hook_tl }
10349 \cs_new_protected:Npn \xunadd_append_end_hook:n
10350 { \tl_put_right:Nn \l__xunadd_end_hook_tl }
10351 \cs_new_protected:Npn \xunadd_set_begin_hook:nn
10352 { \__xunadd_set_cmd_hook:nnn { begin } }
10353 \cs_new_protected:Npn \xunadd_set_end_hook:nn
10354 { \__xunadd_set_cmd_hook:nnn { end } }
10355 \cs_new_protected:Npn \_xunadd_set_cmd_hook:nnn #1#2#3
10356 {
10357   \cs_set_protected:cpn
10358   {
10359     \tl_if_single:nTF {#2}

```



```

10360         { \use:c { __xunadd_#1_csname:n } { \token_to_str:N #2 } }
10361         { \__xunadd_set_cmd_hook_aux:Nnwn #2 \q_stop {#1} }
10362     } ##1
10363     {#3}
10364 }
10365 \cs_new:Npn \__xunadd_set_cmd_hook_aux:Nnwn #1#2 \q_stop #3
10366 { \use:c { __xunadd_#3_csname:n } { \token_to_str:N #1 - \tl_to_str:n {#2} } }
10367 \cs_new:Npn \__xunadd_begin_csname:n #1 { __xunadd_begin_#1_hook:n }
10368 \cs_new:Npn \__xunadd_end_csname:n #1 { __xunadd_end_#1_hook:n }
10369 \tl_new:N \l__xunadd_begin_hook_tl
10370 \tl_new:N \l__xunadd_end_hook_tl

\__xunadd_begin_hook:nn 10371 \cs_new_protected:Npn \__xunadd_begin_hook:nn #1#2
\__xunadd_end_hook:nn 10372 {
10373     \tl_use:N \l__xunadd_begin_hook_tl
10374     \cs_if_exist_use:cF { \__xunadd_begin_csname:n { #1 - \tl_to_str:n {#2} } }
10375     { \cs_if_exist_use:cF { \__xunadd_begin_csname:n {#1} } { \use_none:n } }
10376     {#2}
10377 }
10378 \cs_new_protected:Npn \__xunadd_end_hook:nn #1#2
10379 {
10380     \cs_if_exist_use:cF { \__xunadd_end_csname:n { #1 - \tl_to_str:n {#2} } }
10381     { \cs_if_exist_use:cF { \__xunadd_end_csname:n {#1} } { \use_none:n } }
10382     {#2}
10383     \tl_use:N \l__xunadd_end_hook_tl
10384 }

\DeclareUTFIPACCommand 10385 \NewDocumentCommand \DeclareUTFIPACCommand { 0 { \UTFencname } m }
10386 { \use:e { __xunadd_text_tipa_command:Nnn \exp_not:N #2 { \token_to_str:N #2 } {#1} } }
10387 \cs_new_protected:Npn \__xunadd_text_tipa_command:Nnn #1#2#3
10388 {
10389     \cs_set_eq:cc { UTF/#3#2 } { #3#2 }
10390     \DeclareTextCommand #1 {#3} { \__xunadd_text_tipa_command:nnn {#3} {#2} }
10391 }
10392 \cs_new_protected:Npn \__xunadd_text_tipa_command:nnn #1#2#3
10393 {
10394     \exp_args:Ncc \__xunadd_check_for_tipa:NNn
10395     { \use_none:n #2 } { UTF/#1#2 } {#3}
10396 }
10397 \cs_new_protected:Npn \__xunadd_check_for_tipa:NNn #1#2#3
10398 {
10399     \tl_if_head_eq_meaning:nNTF {#3} \textipa
10400     {
10401         \exp_after:wN \tipacatchonechar \exp_after:wN
10402         { \exp_after:wN #1 \use_none:n #3 }
10403     }
10404     { #2 {#3} }
10405 }

```

\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。

```

10406 \cs_new_protected:Npn \xunadd_get_slot:nn #1#2
10407 { \__xunadd_get_slot:wn #2 \q_nil \q_stop {#1} }
10408 \cs_new_protected:Npn \__xunadd_get_slot:wn #1#2#3 \q_stop #4
10409 {
10410     \int_set:Nn \l_xunadd_slot_int { -1 }
10411     \bool_set_false:N \l_xunadd_rest_bool
10412     \group_begin: \exp_args:Nccc \group_end:
10413     { __xunadd_get_slot:NNnn }
10414     { #4 \token_to_str:N #1 }
10415     { \__xunadd_composite_cs:Nnn #1 {#4} {#2} }
10416     {#2}
10417     {#3}
10418 }
10419 \int_new:N \l_xunadd_slot_int
10420 \bool_new:N \l_xunadd_rest_bool
10421 \cs_new_protected:Npn \__xunadd_get_slot:NNnn #1#2#3#4
10422 {

```

```

10423 \cs_if_free:NF #1
10424 {
10425   \cs_if_exist:NTF #2
10426   { \__xunadd_get_composite_slot:Nn #2 {#4} }
10427   { \__xunadd_get_character_slot:Nn #1 { #3 #4 } }
10428 }
10429 }
10430 \cs_new_protected:Npn \__xunadd_get_composite_slot:Nn #1#2
10431 {
10432   \token_if_chardef:NT #1
10433   {
10434     \int_set:Nn \l_xunadd_slot_int {#1}
10435     \quark_if_nil:nF {#2}
10436     { \bool_set_true:N \l_xunadd_rest_bool }
10437   }
10438 }
10439 \cs_new_protected:Npn \__xunadd_get_character_slot:Nn #1
10440 {
10441   \exp_after:wN \__xunadd_get_character_slot_aux:wn #1
10442   \__xunadd_text_character:nN \q_nil \q_nil \q_stop
10443 }
10444 \cs_new_protected:Npn \__xunadd_get_character_slot_aux:wn
10445 #1 \__xunadd_text_character:nN #2#3#4 \q_stop #5
10446 {
10447   \quark_if_nil:nF {#2}
10448   {
10449     \int_set:Nn \l_xunadd_slot_int {`#3 }
10450     \quark_if_nil:nF {#5}
10451     { \bool_set_true:N \l_xunadd_rest_bool }
10452   }
10453 }

```

\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。

```

10454 \cs_new_protected_nopar:Npn \xunadd@microtype@is@charx #1 \relax
10455 {
10456   \use:e
10457   { \xunadd_get_slot:nn { \MT@encoding } { \tex_the:D \MT@toks } }
10458   \int_compare:nNnTF \l_xunadd_slot_int < \c_zero_int
10459   { \xunadd@original@is@charx #1 \relax }
10460   {
10461     \cs_set_nopar:Npe \MT@char@ { \int_use:N \l_xunadd_slot_int }
10462     \bool_if:NT \l_xunadd_rest_bool { \MT@norestfalse }
10463   }
10464 }
10465 \cs_new_protected_nopar:Npn \xunadd@microtype@hook
10466 {
10467   \cs_if_free:NF \MT@is@charx
10468   {
10469     \cs_new_eq:NN \xunadd@original@is@charx \MT@is@charx
10470     \cs_set_eq:NN \MT@is@charx \xunadd@microtype@is@charx
10471     \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
10472   }
10473 }
10474 \ifpackageloaded { microtype }
10475 { \use:n } { \AtBeginDocument }
10476 { \xunadd@microtype@hook }

10477 </xunicode>
10478 <*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。

```

10479 \DeclareUTFSymbol\texthyphenationpoint{"2027}
10480 \DeclareUTFSymbol\texttwoemdash{"2E3A}

```

以下内容选自 xunicode, 并做了适当修改。

```

10481 \DeclareUTFComposite\textsuperscript
10482 \DeclareUTFComposite\textsubscript
10483 \DeclareUTFEncodedAccent\textsbleftarrow{"20EE}{{"02FF}
10484 \DeclareUTFEncodedAccent\`{"0300}{{"02CB}
10485 \DeclareUTFEncodedAccent\capitalgrave{"0300}{{"02CB}
10486 \DeclareUTFEncodedAccent\'{"0301}{{"02CA}
10487 \DeclareUTFEncodedAccent\capitalacute{"0301}{{"02CA}
10488 \DeclareUTFEncodedAccent\~{"0302}{{"02C6}
10489 \DeclareUTFEncodedAccent\capitalcircumflex{"0302}{{"02C6}
10490 \DeclareUTFEncodedAccent\~{"0303}{{"02DC}
10491 \DeclareUTFEncodedAccent\capitaltilde{"0303}{{"02DC}
10492 \DeclareUTFEncodedAccent\={"0304}{{"02C9}
10493 \DeclareUTFEncodedAccent\capitalmacron{"0304}{{"02C9}
10494 \DeclareUTFEncodedAccent\textoverline{"0305}{{"203E}
10495 \DeclareUTFEncodedAccent\u{"0306}{{"02D8}
10496 \DeclareUTFEncodedAccent\capitalbreve{"0306}{{"02D8}
10497 \DeclareUTFEncodedAccent\..{"0307}{{"02D9}
10498 \DeclareUTFEncodedAccent\capitaldotaccent{"0307}{{"02D9}
10499 \DeclareUTFEncodedAccent\{"0308}{{"00A8}
10500 \DeclareUTFEncodedAccent\capitaldieresis{"0308}{{"00A8}
10501 \DeclareUTFEncodedAccent\m{"0309}{{"0309}
10502 \DeclareUTFEncodedAccent\texthookabove{"0309}{{"0309}
10503 \DeclareUTFEncodedAccent\r{"030A}{{"02DA}
10504 \DeclareUTFEncodedAccent\capitalring{"030A}{{"02DA}
10505 \DeclareUTFEncodedAccent\H{"030B}{{"02DD}
10506 \DeclareUTFEncodedAccent\capitalhungarumlaut{"030B}{{"02DD}
10507 \DeclareUTFEncodedAccent\v{"030C}{{"02C7}
10508 \DeclareUTFEncodedAccent\capitalcaron{"030C}{{"02C7}
10509 \DeclareUTFEncodedAccent\textvbaraccent{"030D}{{"02C8}
10510 \DeclareUTFEncodedAccent\textdoublevbaraccent{"030E}{{"0022}
10511 \DeclareUTFEncodedAccent\U{"030E}{{"0022}
10512 \DeclareUTFEncodedAccent\textdoublegrave{"030F}{{"02F5}
10513 \DeclareUTFEncodedAccent\G{"030F}{{"02F5}
10514 \DeclareUTFEncodedAccent\textdotbreve{"0310}{{"0310}
10515 \DeclareUTFEncodedAccent\textroundcap{"0311}{{"0311}
10516 \DeclareUTFEncodedAccent\newtie{"0311}{{"0311}
10517 \DeclareUTFEncodedAccent\capitalnewtie{"0311}{{"0311}
10518 \DeclareUTFEncodedAccent\textturncommaabove{"0312}{{"02BB}
10519 \DeclareUTFEncodedAccent\textcommaabove{"0313}{{"02BC}
10520 \DeclareUTFEncodedAccent\textrevcommaabove{"0314}{{"02BD}
10521 \DeclareUTFEncodedAccent\overbridge{"0346}{{"0346}
10522 \DeclareUTFEncodedAccent\crtilde{"034A}{{"034A}
10523 \DeclareUTFEncodedAccent\dottedtilde{"034B}{{"034B}
10524 \DeclareUTFEncodedAccent\doubletilde{"034C}{{"034C}
10525 \DeclareUTFEncodedAccent\textrightarrowhead{"0350}{{"02C3}
10526 \DeclareUTFEncodedAccent\textlefthalftring{"0351}{{"02D3}
10527 \DeclareUTFEncodedAccent\texttrighthalftring{"0357}{{"02D2}
10528 \DeclareUTFDoubleEncodedSymbol\textdoublebrevebelow{"035C}{{"035C}
10529 \DeclareUTFDoubleEncodedAccent\textdoublebreve{"035D}{{"035D}
10530 \DeclareUTFDoubleEncodedAccent\textdoublemacron{"035E}{{"035E}
10531 \DeclareUTFDoubleEncodedSymbol\textdoublemacronbelow{"035F}{{"035F}
10532 \DeclareUTFDoubleEncodedAccent\textdoubletilde{"0360}{{"0360}
10533 \DeclareUTFDoubleEncodedAccent\t{"0361}{{"0361}
10534 \DeclareUTFDoubleEncodedAccent\capitaltie{"0361}{{"0361}
10535 \DeclareUTFDoubleEncodedAccent\texttoptiebar{"0361}{{"0361}
10536 \DeclareUTFDoubleEncodedSymbol\sliding{"0362}{{"0362}
10537 \DeclareUTFFTIPACCommand\t
10538 \DeclareUTFFTIPACCommand\capitaltie
10539 \DeclareUTFFTIPACCommand\texttoptiebar
10540 \DeclareUTFFTIPACCommand\sliding
10541 \DeclareUTFEncodedAccent\texthighrise{"1DC4}{{"1DC4}
10542 \DeclareUTFEncodedAccent\textlowrise{"1DC5}{{"1DC5}
10543 \DeclareUTFEncodedAccent\textriseall{"1DC8}{{"1DC8}
10544 \DeclareUTFEncodedAccent\textfallrise{"1DC9}{{"1DC9}
10545 \DeclareUTFEncodedAccent\textaoilig{"1DD5}{{"1DD5}
10546 \DeclareUTFCompositeSymbol\textundertie{H}{{"1E2A}

```

```

10547 \DeclareUTFCompositeSymbol\textundertie{h}{\{"1E2B}
10548 \DeclareUTFEncodedAccents\textcircumgrave{"0302}{\{"0301}
10549 \DeclareUTFSymbol\textFinv{"2132}
10550 \DeclareUTFSymbol\textaleph{"2135}
10551 \DeclareUTFSymbol\textbeth{"2136}
10552 \DeclareUTFSymbol\textgimel{"2137}
10553 \DeclareUTFSymbol\textdaleth{"2138}
10554 \DeclareUTFSymbol\textGame{"2141}
10555 \DeclareUTFCompositeCommand\tonebar{25}{\tonebar{2}\tonebar{5}}
10556 \DeclareUTFCompositeCommand\tonebar{52}{\tonebar{5}\tonebar{2}}
10557 \DeclareUTFSymbol\textbigcircle{"25EF}
10558 \DeclareUTFEncodedCircle\textcircled{"20DD}{\{"25EF}
10559 \DeclareUTFCompositeSymbol\textcircled{0}{\{"24EA}
10560 \DeclareUTFCompositeSymbol\textcircled{1}{\{"2460}
10561 \DeclareUTFCompositeSymbol\textcircled{2}{\{"2461}
10562 \DeclareUTFCompositeSymbol\textcircled{3}{\{"2462}
10563 \DeclareUTFCompositeSymbol\textcircled{4}{\{"2463}
10564 \DeclareUTFCompositeSymbol\textcircled{5}{\{"2464}
10565 \DeclareUTFCompositeSymbol\textcircled{6}{\{"2465}
10566 \DeclareUTFCompositeSymbol\textcircled{7}{\{"2466}
10567 \DeclareUTFCompositeSymbol\textcircled{8}{\{"2467}
10568 \DeclareUTFCompositeSymbol\textcircled{9}{\{"2468}
10569 \DeclareUTFCompositeSymbol\textcircled{10}{\{"2469}
10570 \DeclareUTFCompositeSymbol\textcircled{11}{\{"246A}
10571 \DeclareUTFCompositeSymbol\textcircled{12}{\{"246B}
10572 \DeclareUTFCompositeSymbol\textcircled{13}{\{"246C}
10573 \DeclareUTFCompositeSymbol\textcircled{14}{\{"246D}
10574 \DeclareUTFCompositeSymbol\textcircled{15}{\{"246E}
10575 \DeclareUTFCompositeSymbol\textcircled{16}{\{"246F}
10576 \DeclareUTFCompositeSymbol\textcircled{17}{\{"2470}
10577 \DeclareUTFCompositeSymbol\textcircled{18}{\{"2471}
10578 \DeclareUTFCompositeSymbol\textcircled{19}{\{"2472}
10579 \DeclareUTFCompositeSymbol\textcircled{20}{\{"2473}
10580 \DeclareUTFCompositeSymbol\textcircled{21}{\{"3251}
10581 \DeclareUTFCompositeSymbol\textcircled{22}{\{"3252}
10582 \DeclareUTFCompositeSymbol\textcircled{23}{\{"3253}
10583 \DeclareUTFCompositeSymbol\textcircled{24}{\{"3254}
10584 \DeclareUTFCompositeSymbol\textcircled{25}{\{"3255}
10585 \DeclareUTFCompositeSymbol\textcircled{26}{\{"3256}
10586 \DeclareUTFCompositeSymbol\textcircled{27}{\{"3257}
10587 \DeclareUTFCompositeSymbol\textcircled{28}{\{"3258}
10588 \DeclareUTFCompositeSymbol\textcircled{29}{\{"3259}
10589 \DeclareUTFCompositeSymbol\textcircled{30}{\{"325A}
10590 \DeclareUTFCompositeSymbol\textcircled{31}{\{"325B}
10591 \DeclareUTFCompositeSymbol\textcircled{32}{\{"325C}
10592 \DeclareUTFCompositeSymbol\textcircled{33}{\{"325D}
10593 \DeclareUTFCompositeSymbol\textcircled{34}{\{"325E}
10594 \DeclareUTFCompositeSymbol\textcircled{35}{\{"325F}
10595 \DeclareUTFCompositeSymbol\textcircled{36}{\{"32B1}
10596 \DeclareUTFCompositeSymbol\textcircled{37}{\{"32B2}
10597 \DeclareUTFCompositeSymbol\textcircled{38}{\{"32B3}
10598 \DeclareUTFCompositeSymbol\textcircled{39}{\{"32B4}
10599 \DeclareUTFCompositeSymbol\textcircled{40}{\{"32B5}
10600 \DeclareUTFCompositeSymbol\textcircled{41}{\{"32B6}
10601 \DeclareUTFCompositeSymbol\textcircled{42}{\{"32B7}
10602 \DeclareUTFCompositeSymbol\textcircled{43}{\{"32B8}
10603 \DeclareUTFCompositeSymbol\textcircled{44}{\{"32B9}
10604 \DeclareUTFCompositeSymbol\textcircled{45}{\{"32BA}
10605 \DeclareUTFCompositeSymbol\textcircled{46}{\{"32BB}
10606 \DeclareUTFCompositeSymbol\textcircled{47}{\{"32BC}
10607 \DeclareUTFCompositeSymbol\textcircled{48}{\{"32BD}
10608 \DeclareUTFCompositeSymbol\textcircled{49}{\{"32BE}
10609 \DeclareUTFCompositeSymbol\textcircled{50}{\{"32BF}
10610 \DeclareUTFCompositeSymbol\textcircled{A}{\{"24B6}
10611 \DeclareUTFCompositeSymbol\textcircled{B}{\{"24B7}
10612 \DeclareUTFCompositeSymbol\textcircled{C}{\{"24B8}
10613 \DeclareUTFCompositeSymbol\textcircled{D}{\{"24B9}

```

```

10614 \DeclareUTFCompositeSymbol\textcircled{E}{\textcircled{E}}{"24BA}
10615 \DeclareUTFCompositeSymbol\textcircled{F}{\textcircled{F}}{"24BB}
10616 \DeclareUTFCompositeSymbol\textcircled{G}{\textcircled{G}}{"24BC}
10617 \DeclareUTFCompositeSymbol\textcircled{H}{\textcircled{H}}{"24BD}
10618 \DeclareUTFCompositeSymbol\textcircled{I}{\textcircled{I}}{"24BE}
10619 \DeclareUTFCompositeSymbol\textcircled{J}{\textcircled{J}}{"24BF}
10620 \DeclareUTFCompositeSymbol\textcircled{K}{\textcircled{K}}{"24C0}
10621 \DeclareUTFCompositeSymbol\textcircled{L}{\textcircled{L}}{"24C1}
10622 \DeclareUTFCompositeSymbol\textcircled{M}{\textcircled{M}}{"24C2}
10623 \DeclareUTFCompositeSymbol\textcircled{N}{\textcircled{N}}{"24C3}
10624 \DeclareUTFCompositeSymbol\textcircled{O}{\textcircled{O}}{"24C4}
10625 \DeclareUTFCompositeSymbol\textcircled{P}{\textcircled{P}}{"24C5}
10626 \DeclareUTFCompositeSymbol\textcircled{Q}{\textcircled{Q}}{"24C6}
10627 \DeclareUTFCompositeSymbol\textcircled{R}{\textcircled{R}}{"24C7}
10628 \DeclareUTFCompositeSymbol\textcircled{S}{\textcircled{S}}{"24C8}
10629 \DeclareUTFCompositeSymbol\textcircled{T}{\textcircled{T}}{"24C9}
10630 \DeclareUTFCompositeSymbol\textcircled{U}{\textcircled{U}}{"24CA}
10631 \DeclareUTFCompositeSymbol\textcircled{V}{\textcircled{V}}{"24CB}
10632 \DeclareUTFCompositeSymbol\textcircled{W}{\textcircled{W}}{"24CC}
10633 \DeclareUTFCompositeSymbol\textcircled{X}{\textcircled{X}}{"24CD}
10634 \DeclareUTFCompositeSymbol\textcircled{Y}{\textcircled{Y}}{"24CE}
10635 \DeclareUTFCompositeSymbol\textcircled{Z}{\textcircled{Z}}{"24CF}
10636 \DeclareUTFCompositeSymbol\textcircled{a}{\textcircled{a}}{"24D0}
10637 \DeclareUTFCompositeSymbol\textcircled{b}{\textcircled{b}}{"24D1}
10638 \DeclareUTFCompositeSymbol\textcircled{c}{\textcircled{c}}{"24D2}
10639 \DeclareUTFCompositeSymbol\textcircled{d}{\textcircled{d}}{"24D3}
10640 \DeclareUTFCompositeSymbol\textcircled{e}{\textcircled{e}}{"24D4}
10641 \DeclareUTFCompositeSymbol\textcircled{f}{\textcircled{f}}{"24D5}
10642 \DeclareUTFCompositeSymbol\textcircled{g}{\textcircled{g}}{"24D6}
10643 \DeclareUTFCompositeSymbol\textcircled{h}{\textcircled{h}}{"24D7}
10644 \DeclareUTFCompositeSymbol\textcircled{i}{\textcircled{i}}{"24D8}
10645 \DeclareUTFCompositeSymbol\textcircled{j}{\textcircled{j}}{"24D9}
10646 \DeclareUTFCompositeSymbol\textcircled{k}{\textcircled{k}}{"24DA}
10647 \DeclareUTFCompositeSymbol\textcircled{l}{\textcircled{l}}{"24DB}
10648 \DeclareUTFCompositeSymbol\textcircled{m}{\textcircled{m}}{"24DC}
10649 \DeclareUTFCompositeSymbol\textcircled{n}{\textcircled{n}}{"24DD}
10650 \DeclareUTFCompositeSymbol\textcircled{o}{\textcircled{o}}{"24DE}
10651 \DeclareUTFCompositeSymbol\textcircled{p}{\textcircled{p}}{"24DF}
10652 \DeclareUTFCompositeSymbol\textcircled{q}{\textcircled{q}}{"24E0}
10653 \DeclareUTFCompositeSymbol\textcircled{r}{\textcircled{r}}{"24E1}
10654 \DeclareUTFCompositeSymbol\textcircled{s}{\textcircled{s}}{"24E2}
10655 \DeclareUTFCompositeSymbol\textcircled{t}{\textcircled{t}}{"24E3}
10656 \DeclareUTFCompositeSymbol\textcircled{u}{\textcircled{u}}{"24E4}
10657 \DeclareUTFCompositeSymbol\textcircled{v}{\textcircled{v}}{"24E5}
10658 \DeclareUTFCompositeSymbol\textcircled{w}{\textcircled{w}}{"24E6}
10659 \DeclareUTFCompositeSymbol\textcircled{x}{\textcircled{x}}{"24E7}
10660 \DeclareUTFCompositeSymbol\textcircled{y}{\textcircled{y}}{"24E8}
10661 \DeclareUTFCompositeSymbol\textcircled{z}{\textcircled{z}}{"24E9}
10662 \DeclareUTFCompositeSymbol\textsuperscript{h}{\textsuperscript{h}}{"02B0}
10663 \DeclareUTFCompositeSymbol\textsuperscript{\textth}{\textth}{"02B1}
10664 \DeclareUTFCompositeSymbol\textsuperscript{j}{\textsuperscript{j}}{"02B2}
10665 \DeclareUTFCompositeSymbol\textsuperscript{r}{\textsuperscript{r}}{"02B3}
10666 \DeclareUTFCompositeSymbol\textsuperscript{\textturnr}{\textturnr}{"02B4}
10667 \DeclareUTFCompositeSymbol\textsuperscript{\textturnrrtail}{\textturnrrtail}{"02B5}
10668 \DeclareUTFCompositeSymbol\textsuperscript{\textinvscr}{\textinvscr}{"02B6}
10669 \DeclareUTFCompositeSymbol\textsuperscript{w}{\textsuperscript{w}}{"02B7}
10670 \DeclareUTFCompositeSymbol\textsuperscript{y}{\textsuperscript{y}}{"02B8}
10671 \DeclareUTFCompositeSymbol\textsuperscript{\textbabygamma}{\textbabygamma}{"02E0}
10672 \DeclareUTFCompositeSymbol\textsuperscript{\textgammalatinsmall}{\textgammalatinsmall}{"02E1}
10673 \DeclareUTFCompositeSymbol\textsuperscript{l}{\textsuperscript{l}}{"02E2}
10674 \DeclareUTFCompositeSymbol\textsuperscript{s}{\textsuperscript{s}}{"02E3}
10675 \DeclareUTFCompositeSymbol\textsuperscript{x}{\textsuperscript{x}}{"02E4}
10676 \DeclareUTFCompositeSymbol\textsuperscript{\textrevglotstop}{\textrevglotstop}{"02E5}
10677 \DeclareUTFCompositeSymbol\textsuperscript{\textrelepsilon}{\textrelepsilon}{"1D4C}
10678 \DeclareUTFCompositeSymbol\textsuperscript{\cyrn}{\cyrn}{"1D78}
10679 \DeclareUTFCompositeSymbol\textsuperscript{\textbarsci}{\textbarsci}{"1DA7}
10680 \DeclareUTFCompositeSymbol\textsuperscript{V}{\textsuperscript{V}}{"2C7D}

```

```

10681 \DeclareUTFCompositeSymbol\textsuperscript{\textHbar}{A7F8}
10682 \DeclareUTFCompositeSymbol\textsuperscript{\textHslash}{A7F8}
10683 \DeclareUTFCompositeSymbol\textsuperscript{\oe}{A7F9}
10684 \DeclareUTFCompositeSymbol\textsubscript{h}{2095}
10685 \DeclareUTFCompositeSymbol\textsubscript{k}{2096}
10686 \DeclareUTFCompositeSymbol\textsubscript{l}{2097}
10687 \DeclareUTFCompositeSymbol\textsubscript{m}{2098}
10688 \DeclareUTFCompositeSymbol\textsubscript{n}{2099}
10689 \DeclareUTFCompositeSymbol\textsubscript{p}{209A}
10690 \DeclareUTFCompositeSymbol\textsubscript{s}{209B}
10691 \DeclareUTFCompositeSymbol\textsubscript{t}{209C}

```

以下定义取自 `hyperref` 的 `puenc.def`。

```

10692 \DeclareUTFEncodedAccent\textinvbreve{"0311}{0311}
10693 \DeclareUTFEncodedSymbol\textsubbreve{"032E}{203F}
10694 \DeclareUTFSymbol\textHT{"0009}
10695 \DeclareUTFSymbol\textLF{"000A}
10696 \DeclareUTFSymbol\textCR{"000D}
10697 \DeclareUTFSymbol\textnumbersign{"0023}
10698 \DeclareUTFSymbol\textparenleft{"0028}
10699 \DeclareUTFSymbol\textparenright{"0029}
10700 \DeclareUTFSymbol\textMVPlus{"002B}
10701 \DeclareUTFSymbol\textMVComma{"002C}
10702 \DeclareUTFSymbol\textMVMinus{"002D}
10703 \DeclareUTFSymbol\textMVPeriod{"002E}
10704 \DeclareUTFSymbol\textMVDivision{"002F}
10705 \DeclareUTFSymbol\textMVZero{"0030}
10706 \DeclareUTFSymbol\textMVOne{"0031}
10707 \DeclareUTFSymbol\textMVTwo{"0032}
10708 \DeclareUTFSymbol\textMVThree{"0033}
10709 \DeclareUTFSymbol\textMVFour{"0034}
10710 \DeclareUTFSymbol\textMVFive{"0035}
10711 \DeclareUTFSymbol\textMVSix{"0036}
10712 \DeclareUTFSymbol\textMVSeven{"0037}
10713 \DeclareUTFSymbol\textMVEight{"0038}
10714 \DeclareUTFSymbol\textMVNine{"0039}
10715 \DeclareUTFSymbol\textMVAt{"0040}
10716 \DeclareUTFCompositeCommand\.\{i}{i}
10717 \DeclareUTFCompositeCommand\.\{i}{i}
10718 \DeclareUTFSymbol\textlnot{"00AC}
10719 \DeclareUTFSymbol\textplusminus{"00B1}
10720 \DeclareUTFSymbol\textcedilla{"00B8}
10721 \DeclareUTFSymbol\textmultiply{"00D7}
10722 \DeclareUTFSymbol\textThorn{"00DE}
10723 \DeclareUTFSymbol\textdivide{"00F7}
10724 \DeclareUTFSymbol\textHslash{"0126}
10725 \DeclareUTFCompositeSymbol\k\i{"012F}
10726 \DeclareUTFCompositeSymbol\.\{L}{013F}
10727 \DeclareUTFCompositeSymbol\.\{l}{0140}
10728 \DeclareUTFSymbol\textnapostrophe{"0149}
10729 \DeclareUTFSymbol\textTslash{"0166}
10730 \DeclareUTFSymbol\texttslash{"0167}
10731 \DeclareUTFSymbol\textlongs{"017F}
10732 \DeclareUTFSymbol\texthausaB{"0181}
10733 \DeclareUTFSymbol\texthausaD{"018A}
10734 \DeclareUTFSymbol\textrevE{"018E}
10735 \DeclareUTFSymbol\texthausaK{"0198}
10736 \DeclareUTFSymbol\textPUnrleg{"019E}
10737 \DeclareUTFSymbol\textinve{"01DD}
10738 \DeclareUTFSymbol\textGslash{"01E4}
10739 \DeclareUTFSymbol\textgslash{"01E5}
10740 \DeclareUTFCompositeSymbol\textinvbreve{E}{0206}
10741 \DeclareUTFCompositeSymbol\textinvbreve{e}{0207}
10742 \DeclareUTFCompositeSymbol\textinvbreve{I}{020A}
10743 \DeclareUTFCompositeSymbol\textinvbreve{i}{020B}
10744 \DeclareUTFCompositeSymbol\textinvbreve{\i}{020B}
10745 \DeclareUTFCompositeSymbol\textinvbreve{0}{020E}

```

```

10746 \DeclareUTFCompositeSymbol\textinvbreve{o}{\{"020F}
10747 \DeclareUTFCompositeSymbol\textinvbreve{U}{\{"0216}
10748 \DeclareUTFCompositeSymbol\textinvbreve{u}{\{"0217}
10749 \DeclareUTFSymbol\j{"0237}
10750 \DeclareUTFSymbol\textPUdblig{"0238}
10751 \DeclareUTFSymbol\textPUqplig{"0239}
10752 \DeclareUTFSymbol\textslashc{"023C}
10753 \DeclareUTFSymbol\textniepsilon{"025B}
10754 \DeclareUTFSymbol\textipagamma{"0263}
10755 \DeclareUTFSymbol\textniiota{"0269}
10756 \DeclareUTFSymbol\textnippi{"0278}
10757 \DeclareUTFSymbol\textniupsilon{"028A}
10758 \DeclareUTFSymbol\textring{"02DA}
10759 \DeclareUTFSymbol\texttilde{"02DC}
10760 \DeclareUTFSymbol\texthungarumlaut{"02DD}
10761 \DeclareUTFSymbol\textringlow{"02F3}
10762 \DeclareUTFSymbol\texttildelow{"02F7}
10763 \DeclareUTFCommand\textnewtie{\textinvbreve\ }
10764 \DeclareUTFCommand\textdotbelow{\d\ }
10765 \DeclareUTFSymbol\textmacronbelow{"02CD}
10766 \DeclareUTFCommand\texttie{\t\ }
10767 \DeclareUTFSymbol\textnumeralsigngreek{"0374}
10768 \DeclareUTFSymbol\textnumeralsignlowergreek{"0375}
10769 \DeclareUTFCompositeSymbol\'\{\textAlpha}{\{"0386}
10770 \DeclareUTFCompositeSymbol\'\{\textEpsilon}{\{"0388}
10771 \DeclareUTFCompositeSymbol\'\{\textEta}{\{"0389}
10772 \DeclareUTFCompositeSymbol\'\{\textIota}{\{"038A}
10773 \DeclareUTFCompositeSymbol\'\{\textOmicron}{\{"038C}
10774 \DeclareUTFCompositeSymbol\'\{\textUpsilon}{\{"038E}
10775 \DeclareUTFCompositeSymbol\'\{\textOmega}{\{"038F}
10776 \DeclareUTFCompositeSymbol\'\{\textIotadieresis}{\{"0390}
10777 \DeclareUTFSymbol\textIotadieresis{"03AA}
10778 \DeclareUTFCompositeSymbol\{"\{\textIota}{\{"03AA}
10779 \DeclareUTFCompositeSymbol\{"\{\textUpsilon}{\{"03AB}
10780 \DeclareUTFCompositeSymbol\'\{\textalpha}{\{"03AC}
10781 \DeclareUTFCompositeSymbol\'\{\textepsilon}{\{"03AD}
10782 \DeclareUTFCompositeSymbol\'\{\texteta}{\{"03AE}
10783 \DeclareUTFCompositeSymbol\'\{\textiota}{\{"03AF}
10784 \DeclareUTFCompositeSymbol\{"\{\textupsilonacute}{\{"03B0}
10785 \DeclareUTFSymbol\textmugreek{"03BC}
10786 \DeclareUTFSymbol\textvarsigma{"03C2}
10787 \DeclareUTFCompositeSymbol\{"\{\textiota}{\{"03CA}
10788 \DeclareUTFCompositeSymbol\{"\{\textupsilon}{\{"03CB}
10789 \DeclareUTFCompositeSymbol\'\{\textomicron}{\{"03CC}
10790 \DeclareUTFSymbol\textupsilonacute{"03CD}
10791 \DeclareUTFCompositeSymbol\'\{\textupsilon}{\{"03CD}
10792 \DeclareUTFCompositeSymbol\'\{\textomega}{\{"03CE}
10793 \DeclareUTFSymbol\textStigmatreek{"03DA}
10794 \DeclareUTFSymbol\textstigmatreek{"03DB}
10795 \DeclareUTFSymbol\textDigammagreek{"03DC}
10796 \DeclareUTFSymbol\textdigammagreek{"03DD}
10797 \DeclareUTFSymbol\textKoppagreek{"03DE}
10798 \DeclareUTFSymbol\textkoppagreek{"03DF}
10799 \DeclareUTFSymbol\textSampigreek{"03E0}
10800 \DeclareUTFSymbol\textsampigreek{"03E1}
10801 \DeclareUTFSymbol\textbackepsilon{"03F6}
10802 \DeclareUTFCompositeSymbol\'\{\CYRE}{\{"0400}
10803 \DeclareUTFSymbol\CYRY0{"0401}
10804 \DeclareUTFCompositeSymbol\'\{\CYRE}{\{"0401}
10805 \DeclareUTFSymbol\CYRDJE{"0402}
10806 \DeclareUTFCompositeSymbol\'\{\CYRG}{\{"0403}
10807 \DeclareUTFSymbol\CYRIE{"0404}
10808 \DeclareUTFSymbol\CYRDZE{"0405}
10809 \DeclareUTFSymbol\CYRII{"0406}
10810 \DeclareUTFSymbol\CYRYI{"0407}
10811 \DeclareUTFCompositeSymbol\'\{\CYRII}{\{"0407}
10812 \DeclareUTFSymbol\CYRJE{"0408}

```

```

10813 \DeclareUTFSymbol\CYRLJE{"0409}
10814 \DeclareUTFSymbol\CYRNJE{"040A}
10815 \DeclareUTFSymbol\CYRTSHE{"040B}
10816 \DeclareUTFCompositeSymbol\{'\CYRK}{"040C}
10817 \DeclareUTFCompositeSymbol\{'\CYRI}{"040D}
10818 \DeclareUTFSymbol\CYRUSHRT{"040E}
10819 \DeclareUTFCompositeSymbol\U{\CYRU}{"040E}
10820 \DeclareUTFSymbol\CYRDZHE{"040F}
10821 \DeclareUTFSymbol\CYRA{"0410}
10822 \DeclareUTFSymbol\CYRB{"0411}
10823 \DeclareUTFSymbol\CYRV{"0412}
10824 \DeclareUTFSymbol\CYRG{"0413}
10825 \DeclareUTFSymbol\CYRD{"0414}
10826 \DeclareUTFSymbol\CYRE{"0415}
10827 \DeclareUTFSymbol\CYRZH{"0416}
10828 \DeclareUTFSymbol\CYRZ{"0417}
10829 \DeclareUTFSymbol\CYRI{"0418}
10830 \DeclareUTFSymbol\CYRISHRT{"0419}
10831 \DeclareUTFCompositeSymbol\U{\CYRI}{"0419}
10832 \DeclareUTFSymbol\CYRK{"041A}
10833 \DeclareUTFSymbol\CYRL{"041B}
10834 \DeclareUTFSymbol\CYRM{"041C}
10835 \DeclareUTFSymbol\CYRN{"041D}
10836 \DeclareUTFSymbol\CYRO{"041E}
10837 \DeclareUTFSymbol\CYRP{"041F}
10838 \DeclareUTFSymbol\CYRR{"0420}
10839 \DeclareUTFSymbol\CYRS{"0421}
10840 \DeclareUTFSymbol\CYRT{"0422}
10841 \DeclareUTFSymbol\CYRU{"0423}
10842 \DeclareUTFSymbol\CYRF{"0424}
10843 \DeclareUTFSymbol\CYRH{"0425}
10844 \DeclareUTFSymbol\CYRC{"0426}
10845 \DeclareUTFSymbol\CYRCH{"0427}
10846 \DeclareUTFSymbol\CYRSH{"0428}
10847 \DeclareUTFSymbol\CYRSHCH{"0429}
10848 \DeclareUTFSymbol\CYRHRDSN{"042A}
10849 \DeclareUTFSymbol\CYRERY{"042B}
10850 \DeclareUTFSymbol\CYRSFTSN{"042C}
10851 \DeclareUTFSymbol\CYREREV{"042D}
10852 \DeclareUTFSymbol\CYRYU{"042E}
10853 \DeclareUTFSymbol\CYRYA{"042F}
10854 \DeclareUTFSymbol\cyra{"0430}
10855 \DeclareUTFSymbol\cyrb{"0431}
10856 \DeclareUTFSymbol\cyrv{"0432}
10857 \DeclareUTFSymbol\cyrg{"0433}
10858 \DeclareUTFSymbol\cyrd{"0434}
10859 \DeclareUTFSymbol\cyre{"0435}
10860 \DeclareUTFSymbol\cyrzh{"0436}
10861 \DeclareUTFSymbol\cyrz{"0437}
10862 \DeclareUTFSymbol\cyri{"0438}
10863 \DeclareUTFSymbol\cyrishrt{"0439}
10864 \DeclareUTFCompositeSymbol\U{\cyri}{"0439}
10865 \DeclareUTFSymbol\cyrk{"043A}
10866 \DeclareUTFSymbol\cyr1{"043B}
10867 \DeclareUTFSymbol\cyrm{"043C}
10868 \DeclareUTFSymbol\cyrn{"043D}
10869 \DeclareUTFSymbol\cyro{"043E}
10870 \DeclareUTFSymbol\cyrp{"043F}
10871 \DeclareUTFSymbol\cyrr{"0440}
10872 \DeclareUTFSymbol\cyrs{"0441}
10873 \DeclareUTFSymbol\cyrt{"0442}
10874 \DeclareUTFSymbol\cyru{"0443}
10875 \DeclareUTFSymbol\cyrf{"0444}
10876 \DeclareUTFSymbol\cyrh{"0445}
10877 \DeclareUTFSymbol\cyrc{"0446}
10878 \DeclareUTFSymbol\cyrch{"0447}
10879 \DeclareUTFSymbol\cyrsh{"0448}

```



```

10880 \DeclareUTFSymbol\cyrshch{"0449}
10881 \DeclareUTFSymbol\cyrhrdsn{"044A}
10882 \DeclareUTFSymbol\cyrery{"044B}
10883 \DeclareUTFSymbol\cyrstsn{"044C}
10884 \DeclareUTFSymbol\cyrrrev{"044D}
10885 \DeclareUTFSymbol\cyrYu{"044E}
10886 \DeclareUTFSymbol\cyrYa{"044F}
10887 \DeclareUTFCompositeSymbol\`{\cyre}{"0450}
10888 \DeclareUTFSymbol\cyrYo{"0451}
10889 \DeclareUTFCompositeSymbol\`{\cyre}{"0451}
10890 \DeclareUTFSymbol\cyrdje{"0452}
10891 \DeclareUTFCompositeSymbol\`{\cyrg}{"0453}
10892 \DeclareUTFSymbol\cyrje{"0454}
10893 \DeclareUTFSymbol\cyrdze{"0455}
10894 \DeclareUTFSymbol\cyrri{"0456}
10895 \DeclareUTFSymbol\cyrYi{"0457}
10896 \DeclareUTFCompositeSymbol\`{\cyrri}{"0457}
10897 \DeclareUTFSymbol\cyrje{"0458}
10898 \DeclareUTFSymbol\cyrIje{"0459}
10899 \DeclareUTFSymbol\cyrnje{"045A}
10900 \DeclareUTFSymbol\cyrtshe{"045B}
10901 \DeclareUTFCompositeSymbol\`{\cyrk}{"045C}
10902 \DeclareUTFCompositeSymbol\`{\cyri}{"045D}
10903 \DeclareUTFSymbol\cyrushrt{"045E}
10904 \DeclareUTFCompositeSymbol\U{\curu}{"045E}
10905 \DeclareUTFSymbol\cyrdzhe{"045F}
10906 \DeclareUTFSymbol\CYROMEGA{"0460}
10907 \DeclareUTFSymbol\cyromega{"0461}
10908 \DeclareUTFSymbol\CYRYAT{"0462}
10909 \DeclareUTFSymbol\cyrYat{"0463}
10910 \DeclareUTFSymbol\CYRIOTE{"0464}
10911 \DeclareUTFSymbol\cyrIote{"0465}
10912 \DeclareUTFSymbol\CYRLYUS{"0466}
10913 \DeclareUTFSymbol\cyrlyus{"0467}
10914 \DeclareUTFSymbol\CYRIOTLYUS{"0468}
10915 \DeclareUTFSymbol\cyrIotlyus{"0469}
10916 \DeclareUTFSymbol\CYRBYUS{"046A}
10917 \DeclareUTFSymbol\cyrbyus{"046B}
10918 \DeclareUTFSymbol\CYRIOTBYUS{"046C}
10919 \DeclareUTFSymbol\cyrIotbyus{"046D}
10920 \DeclareUTFSymbol\CYRKSI{"046E}
10921 \DeclareUTFSymbol\cyrksi{"046F}
10922 \DeclareUTFSymbol\CYRPSI{"0470}
10923 \DeclareUTFSymbol\cyrpsi{"0471}
10924 \DeclareUTFSymbol\CYRFITA{"0472}
10925 \DeclareUTFSymbol\cyrfita{"0473}
10926 \DeclareUTFSymbol\CYRIZH{"0474}
10927 \DeclareUTFSymbol\cyrizh{"0475}
10928 \DeclareUTFCompositeSymbol\C{\CYRIZH}{"0476}
10929 \DeclareUTFCompositeSymbol\C{\cyrizh}{"0477}
10930 \DeclareUTFSymbol\CYRUK{"0478}
10931 \DeclareUTFSymbol\cyruk{"0479}
10932 \DeclareUTFSymbol\CYROMEGARND{"047A}
10933 \DeclareUTFSymbol\cyromegarnd{"047B}
10934 \DeclareUTFSymbol\CYROMEGATITLO{"047C}
10935 \DeclareUTFSymbol\cyromegatitlo{"047D}
10936 \DeclareUTFSymbol\CYROT{"047E}
10937 \DeclareUTFSymbol\cyrot{"047F}
10938 \DeclareUTFSymbol\CYRKOPPA{"0480}
10939 \DeclareUTFSymbol\cyrkoppa{"0481}
10940 \DeclareUTFSymbol\cyrthousands{"0482}
10941 \DeclareUTFSymbol\CYRISHRTDSC{"048A}
10942 \DeclareUTFSymbol\cyrishrtdsc{"048B}
10943 \DeclareUTFSymbol\CYRSEMISFTSN{"048C}
10944 \DeclareUTFSymbol\cyrsemisftsn{"048D}
10945 \DeclareUTFSymbol\CYRRTICK{"048E}
10946 \DeclareUTFSymbol\cyrrtick{"048F}

```

```

10947 \DeclareUTFSymbol\CYRGUP{"0490}
10948 \DeclareUTFSymbol\cyrgup{"0491}
10949 \DeclareUTFSymbol\CYRGHCRS{"0492}
10950 \DeclareUTFSymbol\cyrghcrs{"0493}
10951 \DeclareUTFSymbol\CYRGHK{"0494}
10952 \DeclareUTFSymbol\cyrghk{"0495}
10953 \DeclareUTFSymbol\CYRZHDSC{"0496}
10954 \DeclareUTFSymbol\cyrzhdsc{"0497}
10955 \DeclareUTFSymbol\CYRZDSC{"0498}
10956 \DeclareUTFCompositeSymbol\c{\CYRZ}{0498}
10957 \DeclareUTFSymbol\cyrzdsc{"0499}
10958 \DeclareUTFCompositeSymbol\c{\cyrz}{0499}
10959 \DeclareUTFSymbol\CYRKDSC{"049A}
10960 \DeclareUTFSymbol\cyrkdc{"049B}
10961 \DeclareUTFSymbol\CYRKVCRS{"049C}
10962 \DeclareUTFSymbol\cyrkvcrs{"049D}
10963 \DeclareUTFSymbol\CYRKHCRS{"049E}
10964 \DeclareUTFSymbol\cyrkhrs{"049F}
10965 \DeclareUTFSymbol\CYRKBEAK{"04A0}
10966 \DeclareUTFSymbol\cyrkbeak{"04A1}
10967 \DeclareUTFSymbol\CYRNDSC{"04A2}
10968 \DeclareUTFSymbol\cyrndsc{"04A3}
10969 \DeclareUTFSymbol\CYRNG{"04A4}
10970 \DeclareUTFSymbol\cyrng{"04A5}
10971 \DeclareUTFSymbol\CYRPHK{"04A6}
10972 \DeclareUTFSymbol\cyrphk{"04A7}
10973 \DeclareUTFSymbol\CYRABHHA{"04A8}
10974 \DeclareUTFSymbol\cyrabhha{"04A9}
10975 \DeclareUTFSymbol\CYRSDSC{"04AA}
10976 \DeclareUTFCompositeSymbol\k{\CYRS}{04AA}
10977 \DeclareUTFSymbol\cyrsdsc{"04AB}
10978 \DeclareUTFCompositeSymbol\k{\cyrs}{04AB}
10979 \DeclareUTFSymbol\CYRTDSC{"04AC}
10980 \DeclareUTFSymbol\cyrtddsc{"04AD}
10981 \DeclareUTFSymbol\CYRY{"04AE}
10982 \DeclareUTFSymbol\cyry{"04AF}
10983 \DeclareUTFSymbol\CYRYHCRS{"04B0}
10984 \DeclareUTFSymbol\cyryhcrs{"04B1}
10985 \DeclareUTFSymbol\CYRHDSC{"04B2}
10986 \DeclareUTFSymbol\cyrhdsc{"04B3}
10987 \DeclareUTFSymbol\CYRTETSE{"04B4}
10988 \DeclareUTFSymbol\cyrtetse{"04B5}
10989 \DeclareUTFSymbol\CYRCHRDSC{"04B6}
10990 \DeclareUTFSymbol\cyrchrdsc{"04B7}
10991 \DeclareUTFSymbol\CYRCHVCRS{"04B8}
10992 \DeclareUTFSymbol\cyrchvcrs{"04B9}
10993 \DeclareUTFSymbol\CYRSHHA{"04BA}
10994 \DeclareUTFSymbol\cyrshha{"04BB}
10995 \DeclareUTFSymbol\CYRABHCH{"04BC}
10996 \DeclareUTFSymbol\cyrabhch{"04BD}
10997 \DeclareUTFSymbol\CYRABHCHDSC{"04BE}
10998 \DeclareUTFCompositeSymbol\k{\CYRABHCH}{04BE}
10999 \DeclareUTFSymbol\cyrabhchdsc{"04BF}
11000 \DeclareUTFCompositeSymbol\k{\cyrabhch}{04BF}
11001 \DeclareUTFSymbol\CYRpalochka{"04C0}
11002 \DeclareUTFCompositeSymbol\U{\CYRZH}{04C1}
11003 \DeclareUTFCompositeSymbol\U{\cyrz}{04C2}
11004 \DeclareUTFSymbol\CYRKHK{"04C3}
11005 \DeclareUTFSymbol\cyrkhh{"04C4}
11006 \DeclareUTFSymbol\CYRLDSC{"04C5}
11007 \DeclareUTFSymbol\cyrlldsc{"04C6}
11008 \DeclareUTFSymbol\CYRNHK{"04C7}
11009 \DeclareUTFSymbol\cyrnhk{"04C8}
11010 \DeclareUTFSymbol\CYRCHLDSC{"04CB}
11011 \DeclareUTFSymbol\cyrchldsc{"04CC}
11012 \DeclareUTFSymbol\CYRMDSC{"04CD}
11013 \DeclareUTFSymbol\cyrmdsc{"04CE}

```

```

11014 \DeclareUTFCompositeSymbol\U{\CYRA}{04D0}
11015 \DeclareUTFCompositeSymbol\U{\cyra}{04D1}
11016 \DeclareUTFCompositeSymbol\{"\CYRA}{04D2}
11017 \DeclareUTFCompositeSymbol\{"\cyra}{04D3}
11018 \DeclareUTFSymbol\CYRAE{04D4}
11019 \DeclareUTFSymbol\cyrae{04D5}
11020 \DeclareUTFCompositeSymbol\U{\CYRE}{04D6}
11021 \DeclareUTFCompositeSymbol\U{\cyre}{04D7}
11022 \DeclareUTFSymbol\CYRSCHWA{04D8}
11023 \DeclareUTFSymbol\cyrschwa{04D9}
11024 \DeclareUTFCompositeSymbol\{"\CYRSCHWA}{04DA}
11025 \DeclareUTFCompositeSymbol\{"\cyrschwa}{04DB}
11026 \DeclareUTFCompositeSymbol\{"\CYRZH}{04DC}
11027 \DeclareUTFCompositeSymbol\{"\cyrzh}{04DD}
11028 \DeclareUTFCompositeSymbol\{"\CYRZ}{04DE}
11029 \DeclareUTFCompositeSymbol\{"\cyrz}{04DF}
11030 \DeclareUTFSymbol\CYRABHDZE{04E0}
11031 \DeclareUTFSymbol\cyrabhdze{04E1}
11032 \DeclareUTFCompositeSymbol\={\CYRI}{04E2}
11033 \DeclareUTFCompositeSymbol\={\cyri}{04E3}
11034 \DeclareUTFCompositeSymbol\{"\CYRI}{04E4}
11035 \DeclareUTFCompositeSymbol\{"\cyri}{04E5}
11036 \DeclareUTFCompositeSymbol\{"\CYRO}{04E6}
11037 \DeclareUTFCompositeSymbol\{"\cyro}{04E7}
11038 \DeclareUTFSymbol\CYROTLD{04E8}
11039 \DeclareUTFSymbol\cyrotld{04E9}
11040 \DeclareUTFCompositeSymbol\{"\CYROTLD}{04EA}
11041 \DeclareUTFCompositeSymbol\{"\cyrotld}{04EB}
11042 \DeclareUTFCompositeSymbol\{"\CYREREV}{04EC}
11043 \DeclareUTFCompositeSymbol\{"\cyrerev}{04ED}
11044 \DeclareUTFCompositeSymbol\={\CYRU}{04EE}
11045 \DeclareUTFCompositeSymbol\={\cyru}{04EF}
11046 \DeclareUTFCompositeSymbol\{"\CYRU}{04F0}
11047 \DeclareUTFCompositeSymbol\{"\cyru}{04F1}
11048 \DeclareUTFCompositeSymbol\H{\CYRU}{04F2}
11049 \DeclareUTFCompositeSymbol\H{\cyru}{04F3}
11050 \DeclareUTFCompositeSymbol\{"\CYRCH}{04F4}
11051 \DeclareUTFCompositeSymbol\{"\cyrch}{04F5}
11052 \DeclareUTFSymbol\CYRGDSC{04F6}
11053 \DeclareUTFSymbol\cyrgdsc{04F7}
11054 \DeclareUTFCompositeSymbol\{"\CYRERY}{04F8}
11055 \DeclareUTFCompositeSymbol\{"\cyrery}{04F9}
11056 \DeclareUTFSymbol\CYRHHK{04FC}
11057 \DeclareUTFSymbol\cyrhkh{04FD}
11058 \DeclareUTFSymbol\sofpasuq{05C3}
11059 \DeclareUTFSymbol\hebalef{05D0}
11060 \DeclareUTFSymbol\hebbet{05D1}
11061 \DeclareUTFSymbol\hebgimel{05D2}
11062 \DeclareUTFSymbol\hebdalet{05D3}
11063 \DeclareUTFSymbol\hebhe{05D4}
11064 \DeclareUTFSymbol\hebvav{05D5}
11065 \DeclareUTFSymbol\hebzayin{05D6}
11066 \DeclareUTFSymbol\hebbet{05D7}
11067 \DeclareUTFSymbol\hebtet{05D8}
11068 \DeclareUTFSymbol\hebyod{05D9}
11069 \DeclareUTFSymbol\hebfinalkaf{05DA}
11070 \DeclareUTFSymbol\hebkaf{05DB}
11071 \DeclareUTFSymbol\heblamed{05DC}
11072 \DeclareUTFSymbol\hebfinalmem{05DD}
11073 \DeclareUTFSymbol\hebmeme{05DE}
11074 \DeclareUTFSymbol\hebfinalnun{05DF}
11075 \DeclareUTFSymbol\hebnun{05E0}
11076 \DeclareUTFSymbol\hebsamekh{05E1}
11077 \DeclareUTFSymbol\hebayin{05E2}
11078 \DeclareUTFSymbol\hebfinalpe{05E3}
11079 \DeclareUTFSymbol\hebpe{05E4}
11080 \DeclareUTFSymbol\hebfinaltsadi{05E5}

```

```

11081 \DeclareUTFSymbol\hebtsadi{"05E6}
11082 \DeclareUTFSymbol\hebqof{"05E7}
11083 \DeclareUTFSymbol\hebresh{"05E8}
11084 \DeclareUTFSymbol\hebshin{"05E9}
11085 \DeclareUTFSymbol\hebtav{"05EA}
11086 \DeclareUTFSymbol\doublevav{"05F0}
11087 \DeclareUTFSymbol\vavyod{"05F1}
11088 \DeclareUTFSymbol\doubleyod{"05F2}
11089 \DeclareUTFSymbol\textscd{"1D05}
11090 \DeclareUTFSymbol\textPUsck{"1D0B}
11091 \DeclareUTFSymbol\textPUscm{"1D0D}
11092 \DeclareUTFSymbol\textPUscp{"1D18}
11093 \DeclareUTFSymbol\textPUrevscr{"1D19}
11094 \DeclareUTFSymbol\textiinferior{"1D62}
11095 \DeclareUTFSymbol\textrinferior{"1D63}
11096 \DeclareUTFSymbol\textuinferior{"1D64}
11097 \DeclareUTFSymbol\textvinferior{"1D65}
11098 \DeclareUTFSymbol\textbetainferior{"1D66}
11099 \DeclareUTFSymbol\textgammainferior{"1D67}
11100 \DeclareUTFSymbol\textrhoinferior{"1D68}
11101 \DeclareUTFSymbol\textphiinferior{"1D69}
11102 \DeclareUTFSymbol\textchiinferior{"1D6A}
11103 \DeclareUTFSymbol\textbarsci{"1D7B}
11104 \DeclareUTFSymbol\textbarp{"1D7D}
11105 \DeclareUTFSymbol\textbarscu{"1D7E}
11106 \DeclareUTFSymbol\textPurhooka{"1D8F}
11107 \DeclareUTFSymbol\textPurhooke{"1D92}
11108 \DeclareUTFSymbol\textPurhookepsilon{"1D93}
11109 \DeclareUTFSymbol\textPurhookopeno{"1D97}
11110 \DeclareUTFCompositeSymbol\textsubbreve{H}{{"1E2A}
11111 \DeclareUTFCompositeSymbol\textsubbreve{h}{{"1E2B}
11112 \DeclareUTFCompositeSymbol\.\{\textlong}{{"1E9B}
11113 \DeclareUTFSymbol\textcompwordmark{"200C}
11114 \DeclareUTFSymbol\texthdotfor{"2025}
11115 \DeclareUTFSymbol\textprime{"2032}
11116 \DeclareUTFSymbol\textsecond{"2033}
11117 \DeclareUTFSymbol\textthird{"2034}
11118 \DeclareUTFSymbol\textbackprime{"2035}
11119 \DeclareUTFSymbol\textlefttherefore{"2056}
11120 \DeclareUTFSymbol\textfourth{"2057}
11121 \DeclareUTFSymbol\textdiamonddots{"2058}
11122 \DeclareUTFSymbol\textzerosuperior{"2070}
11123 \DeclareUTFSymbol\textisuperior{"2071}
11124 \DeclareUTFSymbol\textfoursuperior{"2074}
11125 \DeclareUTFSymbol\textfivesuperior{"2075}
11126 \DeclareUTFSymbol\textsixsuperior{"2076}
11127 \DeclareUTFSymbol\textsevensuperior{"2077}
11128 \DeclareUTFSymbol\texteightsuperior{"2078}
11129 \DeclareUTFSymbol\textninesuperior{"2079}
11130 \DeclareUTFSymbol\textplussuperior{"207A}
11131 \DeclareUTFSymbol\textminussuperior{"207B}
11132 \DeclareUTFSymbol\textequalssuperior{"207C}
11133 \DeclareUTFSymbol\textparenleftsuperior{"207D}
11134 \DeclareUTFSymbol\textparenrightsuperior{"207E}
11135 \DeclareUTFSymbol\textnsuperior{"207F}
11136 \DeclareUTFSymbol\textzeroinferior{"2080}
11137 \DeclareUTFSymbol\textoneinferior{"2081}
11138 \DeclareUTFSymbol\texttwoinferior{"2082}
11139 \DeclareUTFSymbol\textthreeinferior{"2083}
11140 \DeclareUTFSymbol\textfourinferior{"2084}
11141 \DeclareUTFSymbol\textfiveinferior{"2085}
11142 \DeclareUTFSymbol\textsixinferior{"2086}
11143 \DeclareUTFSymbol\textseveninferior{"2087}
11144 \DeclareUTFSymbol\texteightinferior{"2088}
11145 \DeclareUTFSymbol\textnineinferior{"2089}
11146 \DeclareUTFSymbol\textplusinferior{"208A}
11147 \DeclareUTFSymbol\textminusinferior{"208B}

```

```

11148 \DeclareUTFSymbol\textequalsinferior{"208C}
11149 \DeclareUTFSymbol\textparenleftinferior{"208D}
11150 \DeclareUTFSymbol\textparenrightinferior{"208E}
11151 \DeclareUTFSymbol\textainferior{"2090}
11152 \DeclareUTFSymbol\texteinferior{"2091}
11153 \DeclareUTFSymbol\textoinferior{"2092}
11154 \DeclareUTFSymbol\textxinferior{"2093}
11155 \DeclareUTFSymbol\textschwainferior{"2094}
11156 \DeclareUTFSymbol\texthinferior{"2095}
11157 \DeclareUTFSymbol\textkinferior{"2096}
11158 \DeclareUTFSymbol\textlinferior{"2097}
11159 \DeclareUTFSymbol\textminferior{"2098}
11160 \DeclareUTFSymbol\textninferior{"2099}
11161 \DeclareUTFSymbol\textpinferior{"209A}
11162 \DeclareUTFSymbol\textsinferior{"209B}
11163 \DeclareUTFSymbol\texttinferior{"209C}
11164 \DeclareUTFSymbol\textpeseta{"20A7}
11165 \DeclareUTFSymbol\textDeleatur{"20B0}
11166 \DeclareUTFSymbol\textguarani{"20B2}
11167 \DeclareUTFSymbol\textthslash{"210F}
11168 \DeclareUTFSymbol\textIm{"2111}
11169 \DeclareUTFSymbol\textell{"2113}
11170 \DeclareUTFSymbol\textwp{"2118}
11171 \DeclareUTFSymbol\textRe{"211C}
11172 \DeclareUTFSymbol\textriota{"2129}
11173 \DeclareUTFSymbol\textangstrom{"212B}
11174 \DeclareUTFSymbol\textfax{"213B}
11175 \DeclareUTFSymbol\textinvamp{"214B}
11176 \DeclareUTFSymbol\textoneseventh{"2150}
11177 \DeclareUTFSymbol\textoneninth{"2151}
11178 \DeclareUTFSymbol\textonetenth{"2152}
11179 \DeclareUTFSymbol\textonethird{"2153}
11180 \DeclareUTFSymbol\texttwothirds{"2154}
11181 \DeclareUTFSymbol\textonefifth{"2155}
11182 \DeclareUTFSymbol\texttwofifths{"2156}
11183 \DeclareUTFSymbol\textthreefifths{"2157}
11184 \DeclareUTFSymbol\textfourfifths{"2158}
11185 \DeclareUTFSymbol\textonesixth{"2159}
11186 \DeclareUTFSymbol\textfivesixths{"215A}
11187 \DeclareUTFSymbol\textoneeighth{"215B}
11188 \DeclareUTFSymbol\textthreeeighths{"215C}
11189 \DeclareUTFSymbol\textfiveeighths{"215D}
11190 \DeclareUTFSymbol\textseveeneighths{"215E}
11191 \DeclareUTFSymbol\textrevc{"2184}
11192 \DeclareUTFSymbol\textzerothirds{"2189}
11193 \DeclareUTFSymbol\textnleftarrow{"219A}
11194 \DeclareUTFSymbol\textnrightarrow{"219B}
11195 \DeclareUTFSymbol\texttwoheadleftarrow{"219E}
11196 \DeclareUTFCommand\textntwoheadleftarrow{\textlstrikethru\texttwoheadleftarrow}
11197 \DeclareUTFSymbol\texttwoheaduparrow{"219F}
11198 \DeclareUTFSymbol\texttwoheadrightarrow{"21A0}
11199 \DeclareUTFCommand\textntwoheadrightarrow{\textlstrikethru\texttwoheadrightarrow}
11200 \DeclareUTFSymbol\texttwoheaddownarrow{"21A1}
11201 \DeclareUTFSymbol\textleftarrowtail{"21A2}
11202 \DeclareUTFSymbol\textrightarrowtail{"21A3}
11203 \DeclareUTFSymbol\textmapsto{"21A6}
11204 \DeclareUTFSymbol\texthookleftarrow{"21A9}
11205 \DeclareUTFSymbol\texthookrightarrow{"21AA}
11206 \DeclareUTFSymbol\textlooparrowleft{"21AB}
11207 \DeclareUTFSymbol\textlooparrowright{"21AC}
11208 \DeclareUTFSymbol\textnleftrightarrow{"21AE}
11209 \DeclareUTFSymbol\textlightning{"21AF}
11210 \DeclareUTFSymbol\textdlsb{"21B5}
11211 \DeclareUTFSymbol\textcurvearrowleft{"21B6}
11212 \DeclareUTFSymbol\textcurvearrowright{"21B7}
11213 \DeclareUTFSymbol\textleftharpoonup{"21BC}
11214 \DeclareUTFSymbol\textleftharpoondown{"21BD}

```

```

11215 \DeclareUTFSymbol\textupharpoonright{"21BE}
11216 \DeclareUTFSymbol\textupharpoonleft{"21BF}
11217 \DeclareUTFSymbol\texttrightharpoonup{"21C0}
11218 \DeclareUTFSymbol\texttrightharpoondown{"21C1}
11219 \DeclareUTFSymbol\textdownharpoonright{"21C2}
11220 \DeclareUTFSymbol\textdownharpoonleft{"21C3}
11221 \DeclareUTFSymbol\textrightleftarrows{"21C4}
11222 \DeclareUTFSymbol\textupdownarrows{"21C5}
11223 \DeclareUTFSymbol\textleftrigharrows{"21C6}
11224 \DeclareUTFSymbol\textleftleftarrows{"21C7}
11225 \DeclareUTFSymbol\textupuparrows{"21C8}
11226 \DeclareUTFSymbol\texttrightrightarrows{"21C9}
11227 \DeclareUTFSymbol\textdowndownarrows{"21CA}
11228 \DeclareUTFSymbol\textleftrightharpoons{"21CB}
11229 \DeclareUTFSymbol\textrightleftharpoons{"21CC}
11230 \DeclareUTFSymbol\textnLeftarrow{"21CD}
11231 \DeclareUTFSymbol\textnLeftrightarrow{"21CE}
11232 \DeclareUTFSymbol\textnRrightarrow{"21CF}
11233 \DeclareUTFSymbol\textLeftarrow{"21D0}
11234 \DeclareUTFSymbol\textUparrow{"21D1}
11235 \DeclareUTFSymbol\textRrightarrow{"21D2}
11236 \DeclareUTFSymbol\textDownarrow{"21D3}
11237 \DeclareUTFSymbol\textLeftrightarrow{"21D4}
11238 \DeclareUTFSymbol\textUpdownarrow{"21D5}
11239 \DeclareUTFSymbol\textNrightarrow{"21D6}
11240 \DeclareUTFSymbol\textNearrow{"21D7}
11241 \DeclareUTFSymbol\textSearrow{"21D8}
11242 \DeclareUTFSymbol\textSwarrow{"21D9}
11243 \DeclareUTFSymbol\textLleftarrow{"21DA}
11244 \DeclareUTFSymbol\textRrightarrow{"21DB}
11245 \DeclareUTFSymbol\textleftsquigarrow{"21DC}
11246 \DeclareUTFSymbol\textright squigarrow{"21DD}
11247 \DeclareUTFSymbol\textdashleftarrow{"21E0}
11248 \DeclareUTFSymbol\textdasheduparrow{"21E1}
11249 \DeclareUTFSymbol\textdashrightarrow{"21E2}
11250 \DeclareUTFSymbol\textdasheddownarrow{"21E3}
11251 \DeclareUTFSymbol\textpointer{"21E8}
11252 \DeclareUTFSymbol\textdownuparrows{"21F5}
11253 \DeclareUTFSymbol\textleftarrowtriangle{"21FD}
11254 \DeclareUTFSymbol\textrightarrowtriangle{"21FE}
11255 \DeclareUTFSymbol\textleftrightharpoontriangle{"21FF}
11256 \DeclareUTFSymbol\textforall{"2200}
11257 \DeclareUTFSymbol\textcomplement{"2201}
11258 \DeclareUTFSymbol\textpartial{"2202}
11259 \DeclareUTFSymbol\textexists{"2203}
11260 \DeclareUTFSymbol\textnexists{"2204}
11261 \DeclareUTFSymbol\textemptyset{"2205}
11262 \DeclareUTFSymbol\texttriangle{"2206}
11263 \DeclareUTFSymbol\textnabla{"2207}
11264 \DeclareUTFSymbol\textin{"2208}
11265 \DeclareUTFSymbol\textnotin{"2209}
11266 \DeclareUTFSymbol\textsmallin{"220A}
11267 \DeclareUTFSymbol\textni{"220B}
11268 \DeclareUTFSymbol\textnotowner{"220C}
11269 \DeclareUTFSymbol\textsmallowns{"220D}
11270 \DeclareUTFSymbol\textprod{"220F}
11271 \DeclareUTFSymbol\textamalg{"2210}
11272 \DeclareUTFSymbol\textsum{"2211}
11273 \DeclareUTFSymbol\textmp{"2213}
11274 \DeclareUTFSymbol\textdotplus{"2214}
11275 \DeclareUTFSymbol\textDivides{"2215}
11276 \DeclareUTFSymbol\textsetminus{"2216}
11277 \DeclareUTFSymbol\textast{"2217}
11278 \DeclareUTFSymbol\textcirc{"2218}
11279 \DeclareUTFSymbol\textbulletoperator{"2219}
11280 \DeclareUTFSymbol\textpropto{"221D}
11281 \DeclareUTFSymbol\textinfty{"221E}

```

```

11282 \DeclareUTFSymbol\textangle{"2220}
11283 \DeclareUTFSymbol\textmeasuredangle{"2221}
11284 \DeclareUTFSymbol\textsphericalangle{"2222}
11285 \DeclareUTFSymbol\textmid{"2223}
11286 \DeclareUTFSymbol\textnmid{"2224}
11287 \DeclareUTFSymbol\textparallel{"2225}
11288 \DeclareUTFSymbol\textnparallel{"2226}
11289 \DeclareUTFSymbol\textwedge{"2227}
11290 \DeclareUTFCommand\textowedge{\textcircled\textwedge}
11291 \DeclareUTFSymbol\textvee{"2228}
11292 \DeclareUTFCommand\textovee{\textcircled\textvee}
11293 \DeclareUTFSymbol\textcap{"2229}
11294 \DeclareUTFSymbol\textcup{"222A}
11295 \DeclareUTFSymbol\textint{"222B}
11296 \DeclareUTFSymbol\textiint{"222C}
11297 \DeclareUTFSymbol\textiiint{"222D}
11298 \DeclareUTFSymbol\textoint{"222E}
11299 \DeclareUTFSymbol\textoiint{"222F}
11300 \DeclareUTFSymbol\textointclockwise{"2232}
11301 \DeclareUTFSymbol\textointctrclockwise{"2233}
11302 \DeclareUTFSymbol\texttherefore{"2234}
11303 \DeclareUTFSymbol\textbecause{"2235}
11304 \DeclareUTFSymbol\textvdotdot{"2236}
11305 \DeclareUTFSymbol\textsquaredots{"2237}
11306 \DeclareUTFSymbol\textdotminus{"2238}
11307 \DeclareUTFSymbol\texteqcolon{"2239}
11308 \DeclareUTFSymbol\textsim{"223C}
11309 \DeclareUTFSymbol\textbacksim{"223D}
11310 \DeclareUTFCommand\textnbacksim{\textlstrikethru\textbacksim}
11311 \DeclareUTFSymbol\textwr{"2240}
11312 \DeclareUTFSymbol\textnsim{"2241}
11313 \DeclareUTFSymbol\texteqsim{"2242}
11314 \DeclareUTFCommand\textneqsim{\textlstrikethru\texteqsim}
11315 \DeclareUTFSymbol\textsimeq{"2243}
11316 \DeclareUTFSymbol\textnsimeq{"2244}
11317 \DeclareUTFSymbol\textcong{"2245}
11318 \DeclareUTFSymbol\textncong{"2247}
11319 \DeclareUTFSymbol\textapprox{"2248}
11320 \DeclareUTFSymbol\textnapprox{"2249}
11321 \DeclareUTFSymbol\textapproxeq{"224A}
11322 \DeclareUTFCommand\textnapproxeq{\textlstrikethru\textapproxeq}
11323 \DeclareUTFSymbol\texttriplesim{"224B}
11324 \DeclareUTFCommand\textntriplesim{\textlstrikethru\texttriplesim}
11325 \DeclareUTFSymbol\textbackcong{"224C}
11326 \DeclareUTFCommand\textnbackcong{\textlstrikethru\textbackcong}
11327 \DeclareUTFSymbol\textasympt{"224D}
11328 \DeclareUTFCommand\textnasympt{\textlstrikethru\textasympt}
11329 \DeclareUTFSymbol\textBumpeq{"224E}
11330 \DeclareUTFCommand\textnBumpeq{\textlstrikethru\textBumpeq}
11331 \DeclareUTFSymbol\textbumpeq{"224F}
11332 \DeclareUTFCommand\textnbumpeq{\textlstrikethru\textbumpeq}
11333 \DeclareUTFSymbol\textdoteq{"2250}
11334 \DeclareUTFCommand\textndoteq{\textlstrikethru\textdoteq}
11335 \DeclareUTFSymbol\textdoteqdot{"2251}
11336 \DeclareUTFCommand\textnDoteq{\textlstrikethru\textdoteqdot}
11337 \DeclareUTFSymbol\textfallingdoteq{"2252}
11338 \DeclareUTFCommand\textnfallingdoteq{\textlstrikethru\textfallingdoteq}
11339 \DeclareUTFSymbol\textrisingdoteq{"2253}
11340 \DeclareUTFCommand\textnrisingdoteq{\textlstrikethru\textrisingdoteq}
11341 \DeclareUTFSymbol\textcolonequals{"2254}
11342 \DeclareUTFSymbol\textequalscolon{"2255}
11343 \DeclareUTFSymbol\texteqcirc{"2256}
11344 \DeclareUTFCommand\textneqcirc{\textlstrikethru\texteqcirc}
11345 \DeclareUTFSymbol\textcirceq{"2257}
11346 \DeclareUTFCommand\textncirceq{\textlstrikethru\textcirceq}
11347 \DeclareUTFSymbol\texthateq{"2259}
11348 \DeclareUTFCommand\textnhateq{\textlstrikethru\texthateq}

```

```

11349 \DeclareUTFSymbol\texttriangleeq{"225C}
11350 \DeclareUTFSymbol\textneq{"2260}
11351 \DeclareUTFSymbol\textne{"2260}
11352 \DeclareUTFSymbol\textequiv{"2261}
11353 \DeclareUTFSymbol\textnequiv{"2262}
11354 \DeclareUTFSymbol\textleq{"2264}
11355 \DeclareUTFSymbol\textle{"2264}
11356 \DeclareUTFSymbol\textgeq{"2265}
11357 \DeclareUTFSymbol\textge{"2265}
11358 \DeclareUTFSymbol\textleqq{"2266}
11359 \DeclareUTFCommand\textnleqq{\textlstrikethru\textleqq}
11360 \DeclareUTFSymbol\textgeqq{"2267}
11361 \DeclareUTFCommand\textngeqq{\textlstrikethru\textgeqq}
11362 \DeclareUTFSymbol\textlneqq{"2268}
11363 \DeclareUTFSymbol\textgneqq{"2269}
11364 \DeclareUTFSymbol\textll{"226A}
11365 \DeclareUTFCommand\textnll{\textlstrikethru\textll}
11366 \DeclareUTFSymbol\textgg{"226B}
11367 \DeclareUTFCommand\textngg{\textlstrikethru\textgg}
11368 \DeclareUTFSymbol\textbetween{"226C}
11369 \DeclareUTFSymbol\textnless{"226E}
11370 \DeclareUTFSymbol\textngtr{"226F}
11371 \DeclareUTFSymbol\textnleq{"2270}
11372 \DeclareUTFSymbol\textngeq{"2271}
11373 \DeclareUTFSymbol\textlessssim{"2272}
11374 \DeclareUTFSymbol\textgtrsim{"2273}
11375 \DeclareUTFSymbol\textnlessssim{"2274}
11376 \DeclareUTFSymbol\textngtrsim{"2275}
11377 \DeclareUTFSymbol\textlessgtr{"2276}
11378 \DeclareUTFSymbol\textgtrless{"2277}
11379 \DeclareUTFSymbol\textngtrless{"2278}
11380 \DeclareUTFSymbol\textnlessgtr{"2279}
11381 \DeclareUTFSymbol\textprec{"227A}
11382 \DeclareUTFSymbol\textsucc{"227B}
11383 \DeclareUTFSymbol\textpreccurlyeq{"227C}
11384 \DeclareUTFSymbol\textsucceedcurlyeq{"227D}
11385 \DeclareUTFSymbol\textprecsim{"227E}
11386 \DeclareUTFCommand\textnprecsim{\textlstrikethru\textprecsim}
11387 \DeclareUTFSymbol\textsuccsim{"227F}
11388 \DeclareUTFCommand\textnsuccsim{\textlstrikethru\textsuccsim}
11389 \DeclareUTFSymbol\textnprec{"2280}
11390 \DeclareUTFSymbol\textnsucc{"2281}
11391 \DeclareUTFSymbol\textsubset{"2282}
11392 \DeclareUTFSymbol\textsupset{"2283}
11393 \DeclareUTFSymbol\textnsubset{"2284}
11394 \DeclareUTFSymbol\textnsupset{"2285}
11395 \DeclareUTFSymbol\textsubseteq{"2286}
11396 \DeclareUTFSymbol\textsupseteq{"2287}
11397 \DeclareUTFSymbol\textnsubseteq{"2288}
11398 \DeclareUTFSymbol\textnsupseteq{"2289}
11399 \DeclareUTFSymbol\textsubsetneq{"228A}
11400 \DeclareUTFSymbol\textsupsetneq{"228B}
11401 \DeclareUTFSymbol\textcupdot{"228D}
11402 \DeclareUTFSymbol\textcupplus{"228E}
11403 \DeclareUTFSymbol\textsqsubset{"228F}
11404 \DeclareUTFCommand\textnsqlsubset{\textlstrikethru\textsqsubset}
11405 \DeclareUTFSymbol\textsqsupset{"2290}
11406 \DeclareUTFCommand\textnsqlsupset{\textlstrikethru\textsqsupset}
11407 \DeclareUTFSymbol\textsqsubse{"2291}
11408 \DeclareUTFCommand\textnsqlsubse{"2291}
11409 \DeclareUTFSymbol\textsqsupse{"2292}
11410 \DeclareUTFCommand\textnsqlsupse{"2292}
11411 \DeclareUTFSymbol\textsqcap{"2293}
11412 \DeclareUTFSymbol\textsqcup{"2294}
11413 \DeclareUTFSymbol\textoplus{"2295}
11414 \DeclareUTFSymbol\textominus{"2296}
11415 \DeclareUTFSymbol\textotimes{"2297}

```



```

11416 \DeclareUTFSymbol\textoslash{"2298}
11417 \DeclareUTFSymbol\textodot{"2299}
11418 \DeclareUTFSymbol\textcircledcirc{"229A}
11419 \DeclareUTFSymbol\textcircledast{"229B}
11420 \DeclareUTFSymbol\textcircledash{"229D}
11421 \DeclareUTFSymbol\textboxplus{"229E}
11422 \DeclareUTFSymbol\textboxminus{"229F}
11423 \DeclareUTFSymbol\textboxtimes{"22A0}
11424 \DeclareUTFSymbol\textboxdot{"22A1}
11425 \DeclareUTFSymbol\textvdash{"22A2}
11426 \DeclareUTFSymbol\textdashv{"22A3}
11427 \DeclareUTFCommand\textndashv{\textlstrikethru\textdashv}
11428 \DeclareUTFSymbol\texttop{"22A4}
11429 \DeclareUTFCommand\textndownvdash{\textlstrikethru\texttop}
11430 \DeclareUTFSymbol\textbot{"22A5}
11431 \DeclareUTFCommand\textnupvdash{\textlstrikethru\textbot}
11432 \DeclareUTFSymbol\textvDash{"22A8}
11433 \DeclareUTFSymbol\textVdash{"22A9}
11434 \DeclareUTFSymbol\textVvdash{"22AA}
11435 \DeclareUTFCommand\textnVdash{\textlstrikethru\textVdash}
11436 \DeclareUTFSymbol\textVDash{"22AB}
11437 \DeclareUTFSymbol\textnvdash{"22AC}
11438 \DeclareUTFSymbol\textnvDash{"22AD}
11439 \DeclareUTFSymbol\textnVdash{"22AE}
11440 \DeclareUTFSymbol\textnVDash{"22AF}
11441 \DeclareUTFSymbol\textlhd{"22B2}
11442 \DeclareUTFSymbol\textrrhd{"22B3}
11443 \DeclareUTFSymbol\textunlhd{"22B4}
11444 \DeclareUTFSymbol\textunrhd{"22B5}
11445 \DeclareUTFSymbol\textmultimapdotbothA{"22B6}
11446 \DeclareUTFSymbol\textmultimapdotbothB{"22B7}
11447 \DeclareUTFSymbol\textmultimap{"22B8}
11448 \DeclareUTFSymbol\textveebar{"22BB}
11449 \DeclareUTFSymbol\textbarwedge{"22BC}
11450 \DeclareUTFSymbol\textstar{"22C6}
11451 \DeclareUTFSymbol\textdivideontimes{"22C7}
11452 \DeclareUTFSymbol\textbowtie{"22C8}
11453 \DeclareUTFSymbol\textltimes{"22C9}
11454 \DeclareUTFSymbol\textrtimes{"22CA}
11455 \DeclareUTFSymbol\textleftthreetimes{"22CB}
11456 \DeclareUTFSymbol\textrightthreetimes{"22CC}
11457 \DeclareUTFSymbol\textbacksimeq{"22CD}
11458 \DeclareUTFCommand\textnbacksimeq{\textlstrikethru\textbacksimeq}
11459 \DeclareUTFSymbol\textcurlyvee{"22CE}
11460 \DeclareUTFSymbol\textcurlywedge{"22CF}
11461 \DeclareUTFSymbol\textSubset{"22D0}
11462 \DeclareUTFCommand\textnSubset{\textlstrikethru\textSubset}
11463 \DeclareUTFSymbol\textSupset{"22D1}
11464 \DeclareUTFCommand\textnSupset{\textlstrikethru\textSupset}
11465 \DeclareUTFSymbol\textCap{"22D2}
11466 \DeclareUTFSymbol\textCup{"22D3}
11467 \DeclareUTFSymbol\textpitchfork{"22D4}
11468 \DeclareUTFSymbol\textlessdot{"22D6}
11469 \DeclareUTFSymbol\textgtrdot{"22D7}
11470 \DeclareUTFSymbol\textlll{"22D8}
11471 \DeclareUTFSymbol\textggg{"22D9}
11472 \DeclareUTFSymbol\textlesseqgtr{"22DA}
11473 \DeclareUTFSymbol\textgtreqless{"22DB}
11474 \DeclareUTFSymbol\textcurlyeqprec{"22DE}
11475 \DeclareUTFCommand\textncurlyeqprec{\textlstrikethru\textcurlyeqprec}
11476 \DeclareUTFSymbol\textcurlyeqsucc{"22DF}
11477 \DeclareUTFCommand\textncurlyeqsucc{\textlstrikethru\textcurlyeqsucc}
11478 \DeclareUTFSymbol\textnpreccurlyeq{"22E0}
11479 \DeclareUTFSymbol\textnsucccurlyeq{"22E1}
11480 \DeclareUTFSymbol\textnqsubsetq{"22E2}
11481 \DeclareUTFSymbol\textnqsupsetq{"22E3}
11482 \DeclareUTFSymbol\textsqsubsetq{"22E4}

```

```

11483 \DeclareUTFSymbol\textsqsupseteq{"22E5}
11484 \DeclareUTFSymbol\textlnsim{"22E6}
11485 \DeclareUTFSymbol\textgnsim{"22E7}
11486 \DeclareUTFSymbol\textprecnsim{"22E8}
11487 \DeclareUTFSymbol\textsuccnsim{"22E9}
11488 \DeclareUTFSymbol\textntriangleleft{"22EA}
11489 \DeclareUTFSymbol\textntriangleright{"22EB}
11490 \DeclareUTFSymbol\textntrianglelefteq{"22EC}
11491 \DeclareUTFSymbol\textntrianglerighteq{"22ED}
11492 \DeclareUTFSymbol\textvdots{"22EE}
11493 \DeclareUTFSymbol\textcdots{"22EF}
11494 \DeclareUTFSymbol\textudots{"22F0}
11495 \DeclareUTFSymbol\textddots{"22F1}
11496 \DeclareUTFSymbol\textbarin{"22F6}
11497 \DeclareUTFSymbol\textdiameter{"2300}
11498 \DeclareUTFSymbol\textbackneg{"2310}
11499 \DeclareUTFSymbol\textwasylozenge{"2311}
11500 \DeclareUTFSymbol\textinvbackneg{"2319}
11501 \DeclareUTFSymbol\textclock{"231A}
11502 \DeclareUTFSymbol\textulcorner{"231C}
11503 \DeclareUTFSymbol\texturcorner{"231D}
11504 \DeclareUTFSymbol\textllcorner{"231E}
11505 \DeclareUTFSymbol\textlrcorner{"231F}
11506 \DeclareUTFSymbol\textfrown{"2322}
11507 \DeclareUTFSymbol\textsmile{"2323}
11508 \DeclareUTFSymbol\textKeyboard{"2328}
11509 \DeclareUTFSymbol\textlangle{"2329}
11510 \DeclareUTFSymbol\textrangle{"232A}
11511 \DeclareUTFSymbol\textAPLin{"2339}
11512 \DeclareUTFSymbol\textTumbler{"233C}
11513 \DeclareUTFSymbol\textstmaryrdbaro{"233D}
11514 \DeclareUTFSymbol\textnotslash{"233F}
11515 \DeclareUTFSymbol\textnotbackslash{"2340}
11516 \DeclareUTFSymbol\textboxbackslash{"2342}
11517 \DeclareUTFSymbol\textAPLleftarrowbox{"2347}
11518 \DeclareUTFSymbol\textAPLrightarrowbox{"2348}
11519 \DeclareUTFSymbol\textAPLuparrowbox{"2350}
11520 \DeclareUTFSymbol\textAPLdownarrowbox{"2357}
11521 \DeclareUTFSymbol\textAPLinput{"235E}
11522 \DeclareUTFSymbol\textRequest{"2370}
11523 \DeclareUTFSymbol\textBeam{"2393}
11524 \DeclareUTFSymbol\texthexagon{"2394}
11525 \DeclareUTFSymbol\textAPLbox{"2395}
11526 \DeclareUTFSymbol\textForwardToIndex{"23ED}
11527 \DeclareUTFSymbol\textRewindToIndex{"23EE}
11528 \DeclareUTFSymbol\textbbslash{"244A}
11529 \DeclareUTFSymbol\textCircledA{"24B6}
11530 \DeclareUTFSymbol\textCleaningF{"24BB}
11531 \DeclareUTFCommand\textCleaningFF{\b\textCleaningF}
11532 \DeclareUTFSymbol\textCleaningP{"24C5}
11533 \DeclareUTFCommand\textCleaningPP{\b\textCleaningP}
11534 \DeclareUTFSymbol\textCuttingLine{"2504}
11535 \DeclareUTFSymbol\textUParrow{"25B2}
11536 \DeclareUTFSymbol\textbigtriangleup{"25B3}
11537 \DeclareUTFSymbol\textForward{"25B6}
11538 \DeclareUTFSymbol\texttriangleright{"25B7}
11539 \DeclareUTFSymbol\textRHD{"25BA}
11540 \DeclareUTFSymbol\textDOWNarrow{"25BC}
11541 \DeclareUTFSymbol\textbigtriangledown{"25BD}
11542 \DeclareUTFSymbol\textRewind{"25C0}
11543 \DeclareUTFSymbol\texttriangleleft{"25C1}
11544 \DeclareUTFSymbol\textLHD{"25C4}
11545 \DeclareUTFSymbol\textdiamond{"25C7}
11546 \DeclareUTFSymbol\textlozenge{"25CA}
11547 \DeclareUTFSymbol\textLEFTCIRCLE{"25D6}
11548 \DeclareUTFSymbol\textRIGHTCIRCLE{"25D7}
11549 \DeclareUTFSymbol\textboxbar{"25EB}

```

```

11550 \DeclareUTFSymbol\textCloud{"2601}
11551 \DeclareUTFSymbol\textFiveStar{"2605}
11552 \DeclareUTFSymbol\textFiveStarOpen{"2606}
11553 \DeclareUTFSymbol\textPhone{"260E}
11554 \DeclareUTFSymbol\textboxempty{"2610}
11555 \DeclareUTFSymbol\textCheckedbox{"2611}
11556 \DeclareUTFSymbol\textCrossedbox{"2612}
11557 \DeclareUTFSymbol\textCoffeecup{"2615}
11558 \DeclareUTFSymbol\textHandCuffLeft{"261A}
11559 \DeclareUTFSymbol\textHandCuffRight{"261B}
11560 \DeclareUTFSymbol\textHandLeft{"261C}
11561 \DeclareUTFSymbol\textHandRight{"261E}
11562 \DeclareUTFSymbol\textRadioactivity{"2622}
11563 \DeclareUTFSymbol\textBiohazard{"2623}
11564 \DeclareUTFSymbol\textAnkh{"2625}
11565 \DeclareUTFSymbol\textYinYang{"262F}
11566 \DeclareUTFSymbol\textfrownie{"2639}
11567 \DeclareUTFSymbol\textsmiley{"263A}
11568 \DeclareUTFSymbol\textblacksmiley{"263B}
11569 \DeclareUTFSymbol\textsun{"263C}
11570 \DeclareUTFSymbol\textleftmoon{"263D}
11571 \DeclareUTFSymbol\textrightmoon{"263E}
11572 \DeclareUTFSymbol\textmercury{"263F}
11573 \DeclareUTFSymbol\textPUfemale{"2640}
11574 \DeclareUTFSymbol\textearth{"2641}
11575 \DeclareUTFSymbol\textmale{"2642}
11576 \DeclareUTFSymbol\textjupiter{"2643}
11577 \DeclareUTFSymbol\textsaturn{"2644}
11578 \DeclareUTFSymbol\texturanus{"2645}
11579 \DeclareUTFSymbol\textneptune{"2646}
11580 \DeclareUTFSymbol\textpluto{"2647}
11581 \DeclareUTFSymbol\textaries{"2648}
11582 \DeclareUTFSymbol\texttaurus{"2649}
11583 \DeclareUTFSymbol\textgemini{"264A}
11584 \DeclareUTFSymbol\textcancer{"264B}
11585 \DeclareUTFSymbol\textleo{"264C}
11586 \DeclareUTFSymbol\textvirgo{"264D}
11587 \DeclareUTFSymbol\textlibra{"264E}
11588 \DeclareUTFSymbol\textscorpio{"264F}
11589 \DeclareUTFSymbol\textsagittarius{"2650}
11590 \DeclareUTFSymbol\textcapricornus{"2651}
11591 \DeclareUTFSymbol\textaquarius{"2652}
11592 \DeclareUTFSymbol\textpisces{"2653}
11593 \DeclareUTFSymbol\textspadesuitblack{"2660}
11594 \DeclareUTFSymbol\textheartsuitwhite{"2661}
11595 \DeclareUTFSymbol\textdiamondsuitwhite{"2662}
11596 \DeclareUTFSymbol\textclubsuitblack{"2663}
11597 \DeclareUTFSymbol\textspadesuitwhite{"2664}
11598 \DeclareUTFSymbol\textheartsuitblack{"2665}
11599 \DeclareUTFSymbol\textdiamondsuitblack{"2666}
11600 \DeclareUTFSymbol\textclubsuitwhite{"2667}
11601 \DeclareUTFSymbol\textquaternote{"2669}
11602 \DeclareUTFSymbol\textwonotes{"266B}
11603 \DeclareUTFSymbol\textsixteenthnote{"266C}
11604 \DeclareUTFSymbol\textflat{"266D}
11605 \DeclareUTFSymbol\textnatural{"266E}
11606 \DeclareUTFSymbol\textsharp{"266F}
11607 \DeclareUTFSymbol\textrecycle{"2672}
11608 \DeclareUTFSymbol\textWheelchair{"267F}
11609 \DeclareUTFSymbol\textFlag{"2691}
11610 \DeclareUTFSymbol\textMineSign{"2692}
11611 \DeclareUTFSymbol\textdsilitary{"2694}
11612 \DeclareUTFSymbol\textdsmedical{"2695}
11613 \DeclareUTFSymbol\textdsjuridical{"2696}
11614 \DeclareUTFSymbol\textdschemical{"2697}
11615 \DeclareUTFSymbol\textdsbiological{"2698}
11616 \DeclareUTFSymbol\textdscommercial{"269A}

```

```

11617 \DeclareUTFSymbol\textmanstar{"269D}
11618 \DeclareUTFSymbol\textdanger{"26A0}
11619 \DeclareUTFSymbol\textFemaleFemale{"26A2}
11620 \DeclareUTFSymbol\textMaleMale{"26A3}
11621 \DeclareUTFSymbol\textFemaleMale{"26A4}
11622 \DeclareUTFSymbol\textHermaphrodite{"26A5}
11623 \DeclareUTFSymbol\textNeutral{"26AA}
11624 \DeclareUTFSymbol\textPUuncrfemale{"26B2}
11625 \DeclareUTFSymbol\texthexstar{"26B9}
11626 \DeclareUTFSymbol\textSoccerBall{"26BD}
11627 \DeclareUTFSymbol\textSunCloud{"26C5}
11628 \DeclareUTFSymbol\textRain{"26C6}
11629 \DeclareUTFSymbol\textnoway{"26D4}
11630 \DeclareUTFSymbol\textMountain{"26F0}
11631 \DeclareUTFSymbol\textTent{"26FA}
11632 \DeclareUTFSymbol\textScissorRightBrokenBottom{"2701}
11633 \DeclareUTFSymbol\textScissorRight{"2702}
11634 \DeclareUTFSymbol\textScissorRightBrokenTop{"2703}
11635 \DeclareUTFSymbol\textScissorHollowRight{"2704}
11636 \DeclareUTFSymbol\textPhoneHandset{"2706}
11637 \DeclareUTFSymbol\textTape{"2707}
11638 \DeclareUTFSymbol\textPlane{"2708}
11639 \DeclareUTFSymbol\textEnvelope{"2709}
11640 \DeclareUTFSymbol\textPeace{"270C}
11641 \DeclareUTFSymbol\textWritingHand{"270D}
11642 \DeclareUTFSymbol\textPencilRightDown{"270E}
11643 \DeclareUTFSymbol\textPencilRight{"270F}
11644 \DeclareUTFSymbol\textPencilRightUp{"2710}
11645 \DeclareUTFSymbol\textNibRight{"2711}
11646 \DeclareUTFSymbol\textNibSolidRight{"2712}
11647 \DeclareUTFSymbol\textCheckmark{"2713}
11648 \DeclareUTFSymbol\textCheckmarkBold{"2714}
11649 \DeclareUTFSymbol\textXSolid{"2715}
11650 \DeclareUTFSymbol\textXSolidBold{"2716}
11651 \DeclareUTFSymbol\textXSolidBrush{"2717}
11652 \DeclareUTFSymbol\textPlusOutline{"2719}
11653 \DeclareUTFSymbol\textPlus{"271A}
11654 \DeclareUTFSymbol\textPlusThinCenterOpen{"271B}
11655 \DeclareUTFSymbol\textPlusCenterOpen{"271C}
11656 \DeclareUTFSymbol\textCross{"271D}
11657 \DeclareUTFSymbol\textCrossOpenShadow{"271E}
11658 \DeclareUTFSymbol\textCrossOutline{"271F}
11659 \DeclareUTFSymbol\textCrossMaltese{"2720}
11660 \DeclareUTFSymbol\textDavidStar{"2721}
11661 \DeclareUTFSymbol\textFourAsterisk{"2722}
11662 \DeclareUTFSymbol\textJackStar{"2723}
11663 \DeclareUTFSymbol\textJackStarBold{"2724}
11664 \DeclareUTFSymbol\textClowerTips{"2725}
11665 \DeclareUTFSymbol\textFourStar{"2726}
11666 \DeclareUTFSymbol\textFourStarOpen{"2727}
11667 \DeclareUTFSymbol\textFiveStarOpenCircled{"272A}
11668 \DeclareUTFSymbol\textFiveStarCenterOpen{"272B}
11669 \DeclareUTFSymbol\textFiveStarOpenDotted{"272C}
11670 \DeclareUTFSymbol\textFiveStarOutline{"272D}
11671 \DeclareUTFSymbol\textFiveStarOutlineHeavy{"272E}
11672 \DeclareUTFSymbol\textFiveStarConvex{"272F}
11673 \DeclareUTFSymbol\textFiveStarShadow{"2730}
11674 \DeclareUTFSymbol\textAsteriskBold{"2731}
11675 \DeclareUTFSymbol\textAsteriskCenterOpen{"2732}
11676 \DeclareUTFSymbol\textEightStarTaper{"2734}
11677 \DeclareUTFSymbol\textEightStarConvex{"2735}
11678 \DeclareUTFSymbol\textSixStar{"2736}
11679 \DeclareUTFSymbol\textEightStar{"2737}
11680 \DeclareUTFSymbol\textEightStarBold{"2738}
11681 \DeclareUTFSymbol\textTwelveStar{"2739}
11682 \DeclareUTFSymbol\textSixteenStarLight{"273A}
11683 \DeclareUTFSymbol\textSixFlowerPetalRemoved{"273B}

```

```

11684 \DeclareUTFSymbol\textSixFlowerOpenCenter{"273C}
11685 \DeclareUTFSymbol\textAsterisk{"273D}
11686 \DeclareUTFSymbol\textSixFlowerAlternate{"273E}
11687 \DeclareUTFSymbol\textFiveFlowerPetal{"273F}
11688 \DeclareUTFSymbol\textFiveFlowerOpen{"2740}
11689 \DeclareUTFSymbol\textEightFlowerPetal{"2741}
11690 \DeclareUTFSymbol\textSunshineOpenCircled{"2742}
11691 \DeclareUTFSymbol\textSixFlowerAltPetal{"2743}
11692 \DeclareUTFSymbol\textSnowflakeChevron{"2744}
11693 \DeclareUTFSymbol\textSnowflake{"2745}
11694 \DeclareUTFSymbol\textSnowflakeChevronBold{"2746}
11695 \DeclareUTFSymbol\textSparkle{"2747}
11696 \DeclareUTFSymbol\textSparkleBold{"2748}
11697 \DeclareUTFSymbol\textAsteriskRoundedEnds{"2749}
11698 \DeclareUTFSymbol\textEightFlowerPetalRemoved{"274A}
11699 \DeclareUTFSymbol\textEightAsterisk{"274B}
11700 \DeclareUTFSymbol\textCircleShadow{"274D}
11701 \DeclareUTFSymbol\textSquareShadowBottomRight{"274F}
11702 \DeclareUTFSymbol\textSquareTopRight{"2750}
11703 \DeclareUTFSymbol\textSquareCastShadowBottomRight{"2751}
11704 \DeclareUTFSymbol\textSquareCastShadowTopRight{"2752}
11705 \DeclareUTFSymbol\textDiamondSolid{"2756}
11706 \DeclareUTFSymbol\textRectangleThin{"2758}
11707 \DeclareUTFSymbol\textRectangle{"2759}
11708 \DeclareUTFSymbol\textRectangleBold{"275A}
11709 \DeclareUTFSymbol\textperp{"27C2}
11710 \DeclareUTFCommand\textnotperp{\textlstrikethru\textperp}
11711 \DeclareUTFSymbol\textveedot{"27C7}
11712 \DeclareUTFSymbol\textwedgedot{"27D1}
11713 \DeclareUTFSymbol\textleftspoon{"27DC}
11714 \DeclareUTFSymbol\textlbrackdbl{"27E6}
11715 \DeclareUTFSymbol\textrrackdbl{"27E7}
11716 \DeclareUTFSymbol\textcirclearrowleft{"27F2}
11717 \DeclareUTFSymbol\textcirclearrowright{"27F3}
11718 \DeclareUTFSymbol\textlongleftarrow{"27F5}
11719 \DeclareUTFSymbol\textlongrightarrow{"27F6}
11720 \DeclareUTFSymbol\textlongleftrightarrow{"27F7}
11721 \DeclareUTFSymbol\textLongleftarrow{"27F8}
11722 \DeclareUTFSymbol\textLongrightarrow{"27F9}
11723 \DeclareUTFSymbol\textLongleftrightarrow{"27FA}
11724 \DeclareUTFSymbol\textlongmapsto{"27FC}
11725 \DeclareUTFSymbol\textLongmapsfrom{"27FD}
11726 \DeclareUTFSymbol\textLongmapsto{"27FE}
11727 \DeclareUTFSymbol\textnwsearrow{"2921}
11728 \DeclareUTFSymbol\textneswarrow{"2922}
11729 \DeclareUTFSymbol\textlhooknarrow{"2923}
11730 \DeclareUTFSymbol\textrhoonnearrow{"2924}
11731 \DeclareUTFSymbol\textlhooksearrow{"2925}
11732 \DeclareUTFSymbol\textrhoonswarrow{"2926}
11733 \DeclareUTFSymbol\textleadsto{"2933}
11734 \DeclareUTFSymbol\textrcurvearrowne{"2934}
11735 \DeclareUTFSymbol\textlcurvearrowse{"2935}
11736 \DeclareUTFSymbol\textlcurvearrowsw{"2936}
11737 \DeclareUTFSymbol\textrcurvearrowse{"2937}
11738 \DeclareUTFSymbol\textlcurvearrowdown{"2938}
11739 \DeclareUTFSymbol\textrcurvearrowdown{"2939}
11740 \DeclareUTFSymbol\textrcurvearrowleft{"293A}
11741 \DeclareUTFSymbol\textrcurvearrowright{"293B}
11742 \DeclareUTFSymbol\textleftrightharpoon{"294A}
11743 \DeclareUTFSymbol\textrightleftharpoon{"294B}
11744 \DeclareUTFSymbol\textupdownharpoonrightleft{"294C}
11745 \DeclareUTFSymbol\textupdownharpoonleftright{"294D}
11746 \DeclareUTFSymbol\textleftleftharpoons{"2962}
11747 \DeclareUTFSymbol\textupupharpoons{"2963}
11748 \DeclareUTFSymbol\textrighttrightharpoons{"2964}
11749 \DeclareUTFSymbol\textdowndownharpoons{"2965}
11750 \DeclareUTFSymbol\textleftbarharpoon{"296A}

```

```

11751 \DeclareUTFSymbol\textbarleftharpoon{"296B}
11752 \DeclareUTFSymbol\textbarharpoon{"296C}
11753 \DeclareUTFSymbol\textbarrightharpoon{"296D}
11754 \DeclareUTFSymbol\textupdownharpoons{"296E}
11755 \DeclareUTFSymbol\textdownupharpoons{"296F}
11756 \DeclareUTFSymbol\textllparenthesis{"2987}
11757 \DeclareUTFSymbol\textrrparenthesis{"2988}
11758 \DeclareUTFSymbol\textinvdiameter{"29B0}
11759 \DeclareUTFSymbol\textobar{"29B6}
11760 \DeclareUTFSymbol\textobslash{"29B8}
11761 \DeclareUTFSymbol\textobot{"29BA}
11762 \DeclareUTFSymbol\textNoChemicalCleaning{"29BB}
11763 \DeclareUTFSymbol\textolessthan{"29C0}
11764 \DeclareUTFSymbol\textogreaterthan{"29C1}
11765 \DeclareUTFSymbol\textboxslash{"29C4}
11766 \DeclareUTFSymbol\textboxbslash{"29C5}
11767 \DeclareUTFSymbol\textboxast{"29C6}
11768 \DeclareUTFSymbol\textboxcircle{"29C7}
11769 \DeclareUTFSymbol\textboxbox{"29C8}
11770 \DeclareUTFSymbol\textValve{"29D3}
11771 \DeclareUTFSymbol\textmultimapboth{"29DF}
11772 \DeclareUTFSymbol\textshuffle{"29E2}
11773 \DeclareUTFSymbol\textupplus{"2A04}
11774 \DeclareUTFSymbol\textbigdoublewedge{"2A07}
11775 \DeclareUTFSymbol\textbigdoublevee{"2A08}
11776 \DeclareUTFSymbol\textJoin{"2A1D}
11777 \DeclareUTFSymbol\textfatsemi{"2A1F}
11778 \DeclareUTFSymbol\textcircplus{"2A22}
11779 \DeclareUTFSymbol\textminusdot{"2A2A}
11780 \DeclareUTFSymbol\textdottimes{"2A30}
11781 \DeclareUTFSymbol\textdttimes{"2A32}
11782 \DeclareUTFSymbol\textodiv{"2A38}
11783 \DeclareUTFSymbol\textinvneg{"2A3C}
11784 \DeclareUTFSymbol\textsqdoublecap{"2A4E}
11785 \DeclareUTFSymbol\textcapdot{"2A40}
11786 \DeclareUTFSymbol\textsqdoublecup{"2A4F}
11787 \DeclareUTFSymbol\textdoublewedge{"2A55}
11788 \DeclareUTFSymbol\textdoublevee{"2A56}
11789 \DeclareUTFSymbol\textdoublebarwedge{"2A5E}
11790 \DeclareUTFSymbol\textveedoublebar{"2A63}
11791 \DeclareUTFSymbol\texteqdot{"2A66}
11792 \DeclareUTFCommand\textneqdot{\textlstrikethru\texteqdot}
11793 \DeclareUTFSymbol\textcoloncolonequals{"2A74}
11794 \DeclareUTFSymbol\textleqslant{"2A7D}
11795 \DeclareUTFCommand\textnleqslant{\textlstrikethru\textleqslant}
11796 \DeclareUTFSymbol\textgeqslant{"2A7E}
11797 \DeclareUTFCommand\textngeqslant{\textlstrikethru\textgeqslant}
11798 \DeclareUTFSymbol\textlessapprox{"2A85}
11799 \DeclareUTFCommand\textnlessapprox{\textlstrikethru\textlessapprox}
11800 \DeclareUTFSymbol\textgtrapprox{"2A86}
11801 \DeclareUTFCommand\textngtrapprox{\textlstrikethru\textgtrapprox}
11802 \DeclareUTFSymbol\textlneq{"2A87}
11803 \DeclareUTFSymbol\textgneq{"2A88}
11804 \DeclareUTFSymbol\textlnapprox{"2A89}
11805 \DeclareUTFSymbol\textgnapprox{"2A8A}
11806 \DeclareUTFSymbol\textlesseqqgtr{"2A8B}
11807 \DeclareUTFSymbol\textgtreqqlless{"2A8C}
11808 \DeclareUTFSymbol\texteqslantless{"2A95}
11809 \DeclareUTFSymbol\texteqslantgtr{"2A96}
11810 \DeclareUTFSymbol\textleftslice{"2AA6}
11811 \DeclareUTFSymbol\textright slice{"2AA7}
11812 \DeclareUTFSymbol\textpreceq{"2AAF}
11813 \DeclareUTFCommand\textnpreceq{\textlstrikethru\textpreceq}
11814 \DeclareUTFSymbol\textsucceq{"2AB0}
11815 \DeclareUTFCommand\textnsucceq{\textlstrikethru\textsucceq}
11816 \DeclareUTFSymbol\textprecneq{"2AB1}
11817 \DeclareUTFSymbol\textsuccneq{"2AB2}

```

```

11818 \DeclareUTFSymbol\textpreceqq{"2AB3}
11819 \DeclareUTFCommand\textnpreceqq{\textlstrikethru\textpreceqq}
11820 \DeclareUTFSymbol\textsucceqq{"2AB4}
11821 \DeclareUTFCommand\textnsucceqq{\textlstrikethru\textsucceqq}
11822 \DeclareUTFSymbol\textprecneqq{"2AB5}
11823 \DeclareUTFSymbol\textsuccneqq{"2AB6}
11824 \DeclareUTFSymbol\textprecapprox{"2AB7}
11825 \DeclareUTFCommand\textnprecapprox{\textlstrikethru\textprecapprox}
11826 \DeclareUTFSymbol\textsuccapprox{"2AB8}
11827 \DeclareUTFCommand\textnsuccapprox{\textlstrikethru\textsuccapprox}
11828 \DeclareUTFSymbol\textprecnapprox{"2AB9}
11829 \DeclareUTFSymbol\textsucnapprox{"2ABA}
11830 \DeclareUTFSymbol\textsubseteqq{"2AC5}
11831 \DeclareUTFCommand\textnsubseteqq{\textlstrikethru\textsubseteqq}
11832 \DeclareUTFSymbol\textsupseteqq{"2AC6}
11833 \DeclareUTFCommand\textnsupseteqq{\textlstrikethru\textsupseteqq}
11834 \DeclareUTFSymbol\textdashV{"2AE3}
11835 \DeclareUTFCommand\textndashV{\textlstrikethru\textdashV}
11836 \DeclareUTFSymbol\textDashv{"2AE4}
11837 \DeclareUTFCommand\textnDashv{\textlstrikethru\textDashv}
11838 \DeclareUTFSymbol\textDashV{"2AE5}
11839 \DeclareUTFCommand\textnDashV{\textlstrikethru\textDashV}
11840 \DeclareUTFSymbol\textdownmodels{"2AEA}
11841 \DeclareUTFCommand\textndownmodels{\textlstrikethru\textdownmodels}
11842 \DeclareUTFSymbol\textupmodels{"2AEB}
11843 \DeclareUTFCommand\textnupmodels{\textlstrikethru\textupmodels}
11844 \DeclareUTFSymbol\textupspoon{"2AEF}
11845 \DeclareUTFSymbol\textinterleave{"2AF4}
11846 \DeclareUTFSymbol\textsslash{"2AFD}
11847 \DeclareUTFSymbol\textpentagon{"2B20}
11848 \DeclareUTFSymbol\textvarhexagon{"2B21}
11849 \DeclareUTFSymbol\textjinferior{"2C7C}
11850 \DeclareUTFSymbol\textslashdiv{"2E13}
11851 \DeclareUTFSymbol\textinterrobangdown{"2E18}
11852 \DeclareUTFSymbol\textfivedots{"2E2D}
11853 \DeclareUTFSymbol\textPUheng{"A727}
11854 \DeclareUTFSymbol\textPULhookfour{"A72C}
11855 \DeclareUTFSymbol\textPUscf{"A730}
11856 \DeclareUTFSymbol\textPUaolig{"A735}
11857 \DeclareUTFSymbol\textoo{"A74F}
11858 \DeclareUTFSymbol\textcircumlow{"A788}
11859 \DeclareUTFSymbol\textfi{"FB01}
11860 \DeclareUTFSymbol\textfl{"FB02}
11861 \DeclareUTFSymbol\textGaPa{"1D13B}
11862 \DeclareUTFSymbol\textHaPa{"1D13C}
11863 \DeclareUTFSymbol\textViPa{"1D13D}
11864 \DeclareUTFSymbol\textAcPa{"1D13E}
11865 \DeclareUTFSymbol\textSePa{"1D13F}
11866 \DeclareUTFSymbol\textZwPa{"1D140}
11867 \DeclareUTFSymbol\textfullnote{"1D15D}
11868 \DeclareUTFSymbol\texthalfnote{"1D15E}
11869 \DeclareUTFSymbol\textVier{"1D15F}
11870 \DeclareUTFSymbol\textAcht{"1D160}
11871 \DeclareUTFSymbol\textSech{"1D161}
11872 \DeclareUTFSymbol\textZwdr{"1D162}
11873 \DeclareUTFSymbol\textMundus{"1F30D}
11874 \DeclareUTFSymbol\textMoon{"1F319}
11875 \DeclareUTFSymbol\textManFace{"1F468}
11876 \DeclareUTFSymbol\textWomanFace{"1F469}
11877 \DeclareUTFSymbol\textFax{"1F4E0}
11878 \DeclareUTFSymbol\textFire{"1F525}
11879 \DeclareUTFSymbol\textBicycle{"1F6B2}
11880 \DeclareUTFSymbol\textGentsroom{"1F6B9}
11881 \DeclareUTFSymbol\textLadiesroom{"1F6BA}
11882 \DeclareUTFCommand\textcopyleft{\textcircled\textrevc}
11883 \DeclareUTFCommand\textccsa{\textcircled\textcirclearrowleft}
11884 \DeclareUTFSymbol\textglqq{"201E}

```

```
11885 \DeclareUTFSymbol\textgrqq{"201C}  
11886 \DeclareUTFSymbol\textglq{"201A}  
11887 \DeclareUTFSymbol\textgrq{"2018}  
11888 \DeclareUTFSymbol\textflqq{"00AB}  
11889 \DeclareUTFSymbol\textfrqq{"00BB}  
11890 \DeclareUTFSymbol\textflq{"2039}  
11891 \DeclareUTFSymbol\textfrq{"203A}  
11892 \DeclareUTFSymbol\textneg{"00AC}  
11893 \DeclareUTFSymbol\textcdot{"00B7}  
  
11894 </xunextra>  
11895 <@@=xeCJK>
```

5.22 *xeCJK.cfg*

```
11896 <*config>
```

预设的配置文件 *xeCJK.cfg* 为一个空文件。可以在里面增加设置, 然后保存到本地目录下面。

```
11897
```

```
11898 </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。	150
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。	140	<code>\xeCJK_if_CJK_class</code> : NTF 来代替 <code>\int_case</code> : nnn 判断是否是 CJK 字符类。	102
改用 <code>indentfirst</code> 宏包处理缩进的问题。	146	<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。	132
取消 <code>\cprotect</code> 的外部宏限制。	164	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。	109	增加 Verb 选项。	140
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。	145	<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。	133
<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。	151	<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。	92
<code>_xeCJK_fallback_loop</code> : nnNN: 调整备用字体的循环方式。	121	<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> 。	132
<code>\xeCJK_hook_for_ulem</code> : : 简化对 <code>ulem</code> 宏包的兼容补丁。	165	<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。	91
<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。	129	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。	141	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> 。	140
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。	150	增加小宏包 <code>xeCJK-listings</code> , 用于支持 <code>listings</code> 宏包。	190
增加 <code>NewLineCS</code> 和 <code>EnvCS</code> 选项。	103	<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。	176
增加小宏包 <code>xeCJKintef</code> , 用于处理下划线的问题。	165	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。	98	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。	103	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。	102	解决 <code>CheckSingle</code> 选项与 <code>tablists</code> 宏包的冲突。	102
<code>\xeCJK@family</code> : 修改主要 CJK 字体族的自动更新方式。	150	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。	101	完善对 <code>listings</code> 宏包的支持。	190
<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。	102	<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。	190
<code>\xeCJK_hook_for_ulem</code> : : 完全处理下划线里的标点符号的有关问题。	165	<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> 。	192
<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。	189
<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。	177
<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。	141
<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。	142	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。	164
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。	100	改进获取分区字体属性的办法。	124
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。	102
General: 解决在下划线状态下使用 <code>\makebox</code> 时的错误。	173	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。	150	内部调整分区字体的设置方法。	124
修正重定义 <code>\CJKfamilydefault</code> 无效的问题, 恢复容错能力。	136		

- 使 listings 的 breaklines 选项对 CJK 字符类可用, 并保持标点符号的禁则。..... 193
- 使用 AllowBreakBetweenPuncts 时, 相应标点符号仍能与边界对齐。..... 91
- 修正 xeCJKfntef 与 natbib 等的冲突。..... 165
- 遵循 L^AT_EX3 变量需要预先声明的原则。..... 1
- \addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。..... 134
- CJKfilltwosides: 改用 minipage 和 L^AT_EX 表格 (tabular) 来实现。..... 189
- _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。..... 92
- _xeCJK_fallback_loop:nnNN: 使 \CJKfamilydefault 的 FallBack 设置全局可用。..... 121
- _xeCJK_set_verb_exspace:: 当计算得出的间距为负时, 缩小 CJK 字体。..... 142
- \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
- \xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。..... 27
- v3.2.5** (2013/07/10 – 2013/07/25)
- General: 恢复 \nobreakspace 的原始定义。..... 150
- 解决 fixltx2e 和 amsthm 的冲突。..... 150
- 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
- 增加小宏包 xunicode-addon, 为 xunicode 提供判断字符是否存在的功能。..... 196
- \@setupverbvisiblespace: 可视空格考虑传统 T_EX 字体的情况。..... 143
- Verb: 微调定义。..... 140
- _xeCJK_Boundary_and_FullLeft_glue:N: 细化全角左标点是否位于段首的判断。..... 92
- 增加对 enumitem 宏包修改的 \item 的判断。..... 92
- _xeCJK_boundary_register_text_command:NN: 解决汉字后紧跟 \((...\) 形式的行内数学公式时, 不能加入间距的问题。..... 148
- \xeCJKVerbAddOn: 禁止自动换行, 与西文一致。..... 141
- v3.2.6** (2013/07/29 – 2013/08/15)
- 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。..... 125
- case 类函数的用法与 L^AT_EX3 同步。..... 1
- 为 \mathrm 减少一个可能的数学字体族。..... 148
- \AtEndUTFCommand: 可以指定特定符号命令使用的钩子。..... 206
- _xeCJK_boundary_register_text_command:NN: 考虑 ulem 对 \MakeRobust 的不当定义。..... 148
- 考虑 \math 和 \ensuremath。..... 148
- \xeCJK_CJK_and_Boundary:w: 更好的处理边界是 \relax 的情况。..... 83
- \xeCJK_set_mathfont:: 设置粗体时先检查对应字体是否存在。..... 137
- v3.2.7** (2013/08/22 – 2013/11/09)
- General: 标点宽度设置禁用比例选项的值改为 nan。..... 116
- 处理 AllowBreakBetweenPuncts 与 xeCJKfntef 的兼容问题。..... 91
- 实现自定义行首/尾标点符号宽度功能。..... 105
- 使用 everypage 往 \shipout 盒子里加钩子。..... 24
- 修正 unicode-letters.tex 中谚文符号 \catcode 不准的问题。..... 37
- \Url@MathSetup: 使通过 \UrlFont 等命令设置的 CJK 字体生效。..... 148
- _xeCJK_check_single_aux:nNNw: 与 \CJKspace 兼容。..... 101
- _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。..... 90
- \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。..... 137
- v3.2.8** (2013/11/16 – 2013/12/05)
- General: 启用 xunicode 中的带圈数字和字母设置。..... 208
- \DeclareUTFmathsymbols: 修正 \UseMathAsText 的功能, 恢复 \hbar 和增加以 text 打头的文本符号命令。..... 198
- _xeCJK_nobreak_skip:: 禁止在 \verb 中断行。..... 140
- \xeCJKVerbAddOn: 增加是否是等宽字体的判断。..... 141
- v3.2.9** (2013/12/07 – 2013/12/08)
- General: 文档部分增加 xunicode 定义的符号表。..... 196
- 增加 xunicode-extra.def 中, 用于加入 puenc.def 中的符号定义。..... 212
- \DeclareEncodedCompositeAccents: 修正 xunicode 中的错误定义。..... 202
- \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。..... 154
- v3.2.10** (2014/02/20 – 2014/03/01)
- \CJKaddEncHook: 使用 CJKnumb 时, 让 \Unicode 有定义。..... 164
- \DeclareUTFDoubleEncodedAccent: 改进 \t 等的定义方式。..... 202
- \DeclareUTFDoubleEncodedSymbol: 改进 \sliding 等的定义方式。..... 202
- \DeclareUTFTIPACCommand: 检查 \t 和 \sliding 的参数是否以 \textipa 开头。..... 207
- LoadFandol: 当没有设置字体时, 使用 Fandol 字体系列。..... 135
- v3.2.11** (2014/03/14 – 2014/04/10)
- General: 删除 \xeCJKcaption。..... 164
- 左右角括号 U+2329 和 U+232A 是西文标点符号。..... 31
- \CJK@family: 引入 \CJK@family 保存实际的字体族名。..... 132
- indentfirst: 放弃 indentfirst 和 CJKnumber 选项。..... 145
- \xeCJK_add_to_shipout:n: 不再使用内部名字。..... 24
- v3.2.12** (2014/05/12)
- General: 更新 \int_to_Hex:n。..... 121
- 新增 RubberPunctSkip 选项。..... 106
- v3.2.13** (2014/06/02 – 2014/06/20)
- General: 自动调整 \CJKfamilydefault 时, 只将 \familydefault 展开一次。..... 136
- \xeCJK_set_mathfont:: 修复参数类型错误。..... 137
- v3.2.14** (2014/10/31 – 2014/11/03)
- General: xeCJKfntef 不再依赖 CJKfntef。..... 165
- 解决下划线前后没有 \CJKglue 或 \JKEcglue 的问题。..... 165
- 完善 \varCJKunderline 的实现。..... 165
- v3.2.15** (2014/11/07 – 2014/11/10)
- General: xeCJKfntef 增加 hidden 选项。..... 165
- 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。..... 184

<code>_xeCJK_listings_initial_hook::</code> 修正 breaklines 无效的问题。..... 190 <code>\xeCJKfontofon</code>: 完善选项。..... 181 v3.2.16 (2014/11/20 – 2014/12/16) General: 不再依赖 <code>everypage</code> 宏包。..... 24 修复 <code>\hbar</code>。..... 148 整理 <code>xCJKecglue</code> 的部分代码。..... 82 v3.3.0 (2014/12/26) General: 不把 NS 类中的一些有禁则的日文归入 <code>FullRight</code> 类。..... 31 不把小写日文假名归入 <code>FullRight</code> 类。..... 33 <code>\c_xeCJK_PR_chars_clist</code>: 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 <code>fixltx2e</code> 和 <code>amsthm</code> 的冲突补丁。..... 150 新选项 <code>WidowPenalty</code>。..... 100 <code>\CJKaddEncHook</code>: 应用 0.99992 版的新原语 <code>\Ucharcat</code>。 164 <code>LoadFandol</code>: 为方便 MacTeX 用户, <code>Fandol</code> 字体改用文件名。..... 135 <code>_xeCJK_boundary_register_text_command:NN</code>: 兼容 $\text{\LaTeX}$ 2_ε 2015。..... 148 <code>\xeCJK_check_single_cs:NNn</code>: 补充可能遗漏的空格。 102 <code>\c_xeCJK_CM_chars_clist</code>: 补充音调符号。..... 35 <code>_xeCJK_listings_initial_hook::</code> 解决 <code>prebreak</code> 和 <code>postbreak</code> 功能失效的问题。..... 190 <code>_xeCJK_listings_process_Default:nN</code>: 对 <code>listings</code> 的字符扩展不影响到其符号表中的七位或八位字符。... 192 <code>\xeCJK_token_value_charcode:N</code>: 0.99992 版修复了 <code>\meaning</code> 的 Bug。..... 27 <code>\g_xeCJK_xetex_allocator_int</code>: 兼容 $\text{\LaTeX}$ 2_ε 2015。 152 v3.3.2 (2015/05/15) General: 随 Unicode 7.0 更新简繁体映射。..... 1 <code>\g_xeCJK_xetex_allocator_int</code>: <code>\xe@alloc@intercharclass</code> 总是有定义的。..... 152 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 <code>FullRight</code> 类。..... 31 补充 Ext-E。..... 33 不再把 U+2015 和 U+2500 归入 <code>FullRight</code> 类。..... 31 更新 $\text{\LaTeX}$ X3 代码。..... 1 兼容 $\text{\LaTeX}$ X2_ε 2016/02/01 的字符类设置。..... 30 解决与 <code>microtype</code> 宏包的兼容问题。..... 164 使用新的 Unicode 编码名称 TU。..... 197 <code>CJKfilltwosides</code>: 确保进入水平模式。..... 189 v3.3.4 (2016/02/07) General: 兼容 $\text{\LaTeX}$ X 0.99994 的边界字符类。..... 30 v3.4.0 (2016/05/01 – 2016/05/13) General: <code>RubberPunctSkip</code> 选项有新的值 <code>plus</code> 和 <code>minus</code>。 106 <code>CJKmath</code> 功能也支持分区字体。..... 138 标点符号的压缩量能伸长到原始空白, 能收缩到较小边距。..... 117 改进 <code>xCJKecglue</code> 的实现。..... 45 <code>\xeCJK_set_mathfont::</code> <code>CJKmath</code> 的字符范围遵从 <code>\xeCJKDeclareCharClass</code> 的设置。..... 137	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 <code>textformat</code>。..... 165 修复 <code>CJKspace</code> 功能失效。..... 83 v3.4.2 (2016/10/19) General: 避免在破折号之间折行。..... 40 <code>\xeCJK_clear_Boundary_and_CJK_toks::</code> 提高效率, 避免重复循环。..... 39 v3.4.3 (2016/10/27 – 2016/11/18) <code>\CJKfontspec</code>: 允许字体属性可选项在后的新语法。... 133 <code>\setCJKfallbackfamilyfont</code>: 允许字体属性可选项在后的新语法。..... 123 <code>\setCJKmathfont</code>: 允许字体属性可选项在后的新语法。 137 <code>\setCJKmonofont</code>: 允许字体属性可选项在后的新语法。 133 <code>_xeCJK_long_punct_kerning:N</code>: 考虑破折号边界为负值的情况。..... 111 v3.4.4 (2016/11/30) General: 不压缩长标点与其他标点的间距。..... 116 v3.4.5 (2017/01/02) General: 更新 $\text{\LaTeX}$ X3 的过时用法。..... 148 v3.4.6 (2017/02/23) <code>\xeCJK@family</code>: 将族名参数完全展开, 以解决与 <code>fontspec</code> 2017/01/24 v2.5d 的兼容问题。..... 150 v3.4.7 (2017/03/20) General: 简化 <code>CheckSingle</code> 的实现, 不再展开宏。..... 101 v3.4.8 (2017/05/15) General: 转义 <code>\lstinline</code> 参数中的 <code>_12</code>。..... 196 v3.5.0 (2017/07/19 – 2017/07/22) General: 补充 Ext-F。..... 33 常数 <code>\c_minus_one</code> 已过时。..... 1 使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 ($\text{\LaTeX}$ X3 2017/07/19)。..... 1 v3.5.1 (2017/11/16) General: 修正 <code>fallback</code> 字体后无法忽略空格的错误。... 121 v3.6.0 (2018/01/13 – 2018/01/24) General: <code>Default</code> 类与 <code>MiddlePunct</code> 之间不应该有 <code>\CJKglue</code>。..... 97 把 TWO-EM DASH (U+2E3A) 归入 <code>FullRight</code> 类和设为 <code>LongPunct</code> 与 <code>MiddlePunct</code>。..... 31 将全角浪线 U+FF5E 等连接号归入 <code>FullRight</code> 类和设为 <code>MiddlePunct</code>。..... 31 解决标点中间被隔开的禁则与压缩问题。..... 95 同步 $\text{\LaTeX}$ X3 2017/12/16。..... 1 新增 <code>PunctFamily</code> 选项支持对汉字标点单独切换字体。 131 修正标点同为 <code>LongPunct</code> 与 <code>MiddlePunct</code> 时的实现错误。..... 97 总允许长标点与其他标点之间折行。..... 90 v3.6.1 (2018/02/25 – 2018/02/27) General: 减少 <code>bool</code> 运算。..... 1 <code>\xeCJK_if_last_punct:TF</code>: 细化判断。..... 95 v3.7.0 (2018/03/12 – 2018/03/18) General: 补充定义 <code>\textthyphenationpoint</code> 和 <code>\texttwoemdash</code>。..... 208 不再默认引入 <code>xunicode</code> 宏包。..... 146 对 <code>\nobreakspace</code> 的恢复放到 <code>xunicode-addon</code> 中处理。..... 150
--	--

修正长标点被隔开时的压缩处理错误。.....	111	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。...	206	\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。.....	189	同步 Unicode 14.0。.....	33
简化 CJKspace 的实现, 并修复错误。.....	83	(2022/05/26)	
解决与 microtype 宏包的兼容问题。.....	208	v3.8.9	
删除定义新字体族时过滤重复选项的功能。.....	126	General: 修正居中标点悬挂错误。.....	115
同步 L ^A T _E X3 2019/03/05。.....	1	_xeCJK_bound_type_1_glue:Nn: 增加位于段首的支架	
\xeCJK_FullLeft_and_Default:: 再次修正 FullLeft 类		盒子判断。.....	93
字符与西文连用断词失败的问题。.....	91	v3.9.0 (2022/06/06 – 2022/07/08)	
_xeCJK_patch_tuenc_composite:: 修复补丁错误。..	154	General: 不直接依赖 xparse 和 l3keys2e。.....	23
v3.7.3 (2019/04/15)		修复西文的 character protrusion 功能。.....	82
General: 补充日文假名扩展。.....	33	v3.9.1 (2022/07/28 – 2022/08/02)	
修复 penalty 数值错误。.....	95	General: 简化部分内部实现。.....	49
v3.7.4 (2019/05/31)		修复下划线中数学公式的错误处理。.....	174
General: 简化行首/尾标点符号宽度的实现。.....	105	v3.10.0 (2022/08/31 – 2026/06/23)	
v3.8.0 (2020/02/09 – 2020/02/10)		General: NoBreakLongPunct 在右侧时禁止折行。.....	90
General: 兼容 L ^A T _E X 2 _ε 2020/02/02 对 NFSS 的修改。.....	1	\reset@color 补丁增加 hlist 回退路径, 当 color pop 后	
清理过时的兼容性补丁代码。.....	1	最后节点为 hlist 时设置	
删除 CJKfntef 补丁。.....	164	\g__xeCJK_reset_color_pending_bool, 延迟到	
删除 realscripts 补丁。.....	150	_xeCJK_check_for_glue_skip: 中处理 \colorbox	
删除 \hbar 补丁。.....	148	等命令右侧间距(#831)。.....	159
删除 \mathrm 补丁。.....	148	补丁 l3color 后端的 _color_select:N 和	
应用 \fp_if_nan:nTF。.....	116	_color_backend_reset:, 使 l3color 接口的颜色切换	
应用 \peek_remove_spaces:n。.....	27	也能正确保持 xeCJK 间距(#832)。.....	160
\@setupverbvisiblespace: 更新可视空格补丁。.....	143	补丁 \reset@color 以在颜色弹出 whatsit 之后重新放	
v3.8.1 (2020/02/14)		置 xeCJK 节点标记, 修复 \textcolor 命令右侧多余	
General: 修复 \l_xeCJK_current_font_tl 标记错误。..	128	inter-word glue 的问题(#831)。.....	159
应用 \shapedefault。.....	138	撤回对 \char 的重定义, 改为提供 \xeCJKchar 新命令,	
v3.8.2 (2020/02/17)		避免破坏依赖	
General: 避免导言区字体警告。.....	133	let	
修复分区字体错误。.....	129	保存 \tn {char} 原语的宏包(\#800)。hdclindex8477155	
v3.8.3 (2020/03/15 – 2020/04/27)		改正拼写错误 _xeCJK_glue_node:n。.....	49
General: hidden 选项保留原内容的高度和深度。.....	172	将全角浪线等连接号从 MiddlePunct 改为 LongPunct,	
补充 U+02EA 和 U+02EB。.....	33	消除不必要的标点压缩间距。.....	31
兼容 unicode-math 和 CJKmath 选项。.....	156	使用 \l_keys_key_str 和 \l_keys_choice_str 替代	
取消 xeCJKfntef 的初始彩色设置。.....	187	已废弃的 _tl 版本(#806)。.....	159
删除 nopar。.....	1	提升 L ^A T _E X3 最低版本要求至 2025/10/09。.....	22
同步 Unicode 13.0。.....	33	提升版本号至 v3.10.0。.....	1
修复 xCJKecglue 选项。.....	95	通过在 \Hy@BeginAnnot 中保存并选择性恢复 xeCJK 节	
依赖 ctexhook 宏包。.....	24	点标记, 同时清除旧标记, 解决目录中链接注释起始处的	
重构 PunctStyle 选项, 完全展开参数。.....	119	虚假 ecglue(#810), 并为 \ref 提供前侧 ecglue(#809)。157	
v3.8.4 (2020/05/31 – 2020/06/04)		同步 Unicode 15.0。.....	33
General: 重构后备字体的实现, 修正标点符号无后备字		同步 Unicode 15.1。.....	33
体的问题。.....	120	同步 Unicode 17.0。.....	33
\xeCJK_font_gset_to_current:N: 不缓存 \nullfont。..	25	为 color/xcolor 添加兼容补丁, 在颜色切换 whatsit 后重	
v3.8.5 (2020/06/25 – 2020/06/26)		放 xeCJK 节点标记, 修复 \textcolor 后 CJKecglue 丢	
General: 进一步兼容 microtype。.....	156	失的问题(#315, #803)。.....	159
_xeCJK_bound_type_1_glue:Nn: 增加盒子高度判断。..	93	为 hypdoc 的 \HD@target 添加补丁, 在它产生的 hbox	
v3.8.6 (2020/10/17 – 2020/10/18)		之后重放 xeCJK 节点标记, 修复 l3doc 中 \cs、\meta 等	
General: 兼容 L ^A T _E X 2020/10/01 的 NFSS 钩子机制。...	150	命令后 CJKecglue 丢失或保留为原始空格的问题	
正确还原标点符号后的 penalty 状态。.....	95	(#873)。.....	160
v3.8.7 (2021/06/04 – 2021/06/09)		为 mtp ₂ 提供兼容补丁, 使大花括号内部的 \char 不	
General: 更好地兼容 CJKnumb。.....	164	被 interchar 拦截(#407)。.....	156

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)。	161
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)。	106	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)。	157
新增 PoZheHaoLigature 选项, 支持字体的破折号合字功能(#382)。	106	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 l3color 的颜色 push/pop 入口注册为 transparent capture, 删除对应专用 marker 保护分支(#992)。	160
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)。	90	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)。	159
_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)。	111	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)。	113	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)。	113	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)。	160
\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)。	161
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::	
开明句式末点号的宽度由 0.8em 改为全角(#975)。	120	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”。	
全角格式启用相邻标点边界优化, 并保留左标点右接右标点的自然空白(#975)。	120	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
提升版本号至 v3.10.3。	1	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
新增 CJLineBreak 选项, 为日文小假名提供可选严格行首禁则(#165)。	106	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 CJStarter 类和可选严格行首禁则, 保持小假名的普通 CJK 字距(#165)。	41	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 enabled-left-right-kerning 标点格式选项(#975)。	14	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
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)。	157	\xeCJK_check_for_glue:: Boundary 到 CJK 的源码空格检查改用与 Default 方向相同的	
\CJKsout: 使用 \cleaders 使删除线相对正文居中(#967)。	183	_xeCJK_skip_if_interword:N 判断函数, 自然宽度不等于当前词间空格的显式 glue 不再被替换为 \CJKglue(#996)。	46
\CJKKunderanyline: 使用 \cleaders 使自定义线相对正文居中(#967)。	184	由命令边界 framework 接管 xeCJKfntef、盒子命令和颜	
\CJKKunderdblline: 使用 \cleaders 使双下划线相对正文居中(#967)。	183		
\CJKKunderline: 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)。	182		
\CJKKunderwave: 使用 \xleaders 使波浪线相对正文对齐, 同时保持字间连接连续(#967)。	183		
\CJKxout: 使用 \cleaders 使斜删除线相对正文居中(#967)。	184		
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)。	164		
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)。	160		

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)。	185
_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)。	195
\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)。	180
让嵌套的 ulem/xecjkfntef 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)。	180
v3.10.5 (2026/07/25 – 2026/08/05)	
General: 把 l3doc 的 stream capture 从内层 _codedoc_meta:n 上移到公开的 \meta, 使左右两侧的 \CJKecglue 都按命令外的字体度量求值(#1046)。	162
把行内锚点的 \hyper@anchor、\Hy@raisedlink 与 _hyp_target_raise:n 三个出口注册为 transparent capture, 使 \hypertarget、\phantomsection、\MakeLinkTarget、无编号标题锚点与手工包裹的锚点不再遮蔽其前后的实际边界; 经 \hyper@anchorstart 裸调用的路径(如 \pdfbookmark)尚未覆盖(#1047)。	157
_xeCJK_boundary_replay_before:: 新增按花括号转发参数的 transparent 包装变体, 供把参数再次用作 \hbox:n 内容的目标函数使用(#1047)。	80
\xeCJK_CJK_and_Boundary:w: 只吸收触发转换的那一枚左花括号, 不再吞掉整个分组再重新发出, 修复 tabular 中中文\报 Improper alphabetic constant 的问题(#1038)。	84
\l_xeCJK_fntef_pattern_dim: 波浪线和斜删除线改用随字号缩放的四分之一字宽绘图单元和普通 \leaders; 带减号形式在首尾各内缩半个周期, 避免相邻命令连在一起(#1012)。	182
\xeCJK_fntef_sbox:n: 用空的 ActualText 标记装饰内容, 并在构造装饰盒时暂停 LaTeX tagging, 避免波浪下划线和斜删除线使用的字形进入 PDF 文本提取结果(#1017)。	185
_xeCJK_skip_if_interword:N: \CJKecglue 补在源码空格位置时改用可搬运通道, 避免收缩量固化在 ulem 的定宽片段盒内(#1037)。	43
\xeCJK_ulem_on:n: 在用户手册说明线型命令相互嵌套时被嵌套的一段无法断行, 并区分线型与符号型命令、给出替代写法; 同时以「能否断行」为可观察量给 \UL@onin 路径补上回归保护(#1057)。	180
正文一律以字面记号进入 ulem 的参数: 只有“公式加尾随源码空格”才重排, 且重排也先拼装再一次展开, 避免西文词右侧的 \CJKecglue 被关进固定宽度装饰片段盒而丢失收缩量(#1026)。	180
_xeCJK_ulem_periodic_right_skip:: 周期装饰按嵌套层保存末段状态, 避免内层命令改写外层右端(#1012)。	171
_xeCJK_ulem_punct_ccglue:: \CJKecglue 补在源码空格位置时改用可搬运通道, 避免收缩量固化在 ulem 的定宽片段盒内(#1037)。	179

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

\#	1430, 1440, 3233, 3234, 3271, 3275
\&	1466, 1474, 1479
\(	1512
\)	1550
\\	5, 6, 7, 13, 14, 15, 16, 31, 366, 367
\}	1440
\(font-switch)	9

A

\addCJKfontfeatures	10
\AddToHook	2847, 2849, 2860, 2862, 2873, 2875, 2886, 2896, 2898
AllowBreakBetweenPuncts	6
\AtBeginDocument	70
\AtEndOfPackage	433, 729, 742, 753, 777, 807
AutoFakeBold	4, 9
AutoFakeSlant	4, 9
AutoFallBack	4

B

\begin	1449
bool commands:	
\bool_gset_false:N	931, 947, 959, 1095, 1105, 1134, 1151, 1777, 1892, 2131, 2219, 2361, 2464, 2594
\bool_gset_true:N	1776, 2839, 3104, 3430
\bool_if:NTF	102, 320, 610, 612, 633, 904, 946, 1082, 1092, 1096, 1103, 1111, 1121, 1133, 1137, 1150, 1182, 1763, 2066, 2069, 2331, 2949, 2968, 2970, 3052, 3062, 3406, 3450
\bool_lazy_and:nnTF	2378
\bool_lazy_or:nnTF	557
\bool_lazy_or_p:nn	2384
\bool_new:N	53, 111, 270, 615, 616, 637, 727, 1311, 1319, 1320, 1321, 3378, 3398
\bool_set_false:N	245, 2952, 2953, 3352, 3390
\bool_set_true:N	104, 252, 723, 2961, 2966, 2984, 2990, 3008, 3014, 3024, 3037, 3046, 3056, 3066, 3341, 3363, 3384
Boundary	369
box commands:	
\box_clear:N	2409, 2446
\box_dp:N	1949, 2391
\box_ht:N	1948, 2387
\box_if_exist:NTF	1338
\box_new:N	51, 1327, 1328, 1329, 1330, 1339, 2368
\box_set_to_last:N	1944, 2355
\box_use_drop:N	1953, 1981, 1985, 2003, 2011, 2015, 2510, 2527
\box_wd:N	224, 1947, 2381
boxdepth	18

C

\catcode	1466, 1474
----------	------------

\changes	1428, 3232, 3269
----------	------------------

char commands:

\char_generate:nn	436
\char_set_catcode_alignment:N	1479
\char_set_catcode_ignore:n	332, 333, 334, 335, 336
\char_set_catcode_letter:n	575
CheckFullRight	6
CheckSingle	4
CJK	369
\CJKecglue	841, 1006, 1400, 1404, 2046, 2053, 3452
CJKecglue	3, 3331
\CJKfamily	9
\CJKfamilydefault	10
CJKfilltwosides	19
\CJKfontspec	10
\CJKglue	805, 1113, 1123, 1159, 1163, 1185, 1228, 2046, 2057, 3326, 3392
CJKglue	3, 3322
CJKmath	3
\CJKrmdefault	10
\CJKsfdefault	10
\CJKsout	17
CJKspace	3, 3379
\CJKsymbol	796, 828, 1047
\CJKttdefault	10
\CJKkunderanyline	18
\CJKkunderanysymbol	18
\CJKkunderdblline	17
\CJKkunderdot	17
\CJKkunderline	17
\CJKkunderwave	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	2543, 2605, 2640, 2673, 2732, 2735, 2794, 2820, 2833
\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, 990, 3241
\clist_new:N	55, 346, 358
\clist_set:Nn	535
CM	388

\cs .. 1429, 1432, 1433, 1437, 1440, 1445, 1450, 1452, 1453,
1454, 1456, 1457, 1459, 1462, 1466, 1468, 1469, 1475,
1476, 1536, 1539, 2903, 3280, 3286, 3287, 3289, 3290, 3291

cs commands:

\cs_end: 308
 \cs_generate_variant:Nn 554, 644, 657, 663, 2692
 \cs_gset_eq:NN 131, 142, 3305
 \cs_gset_protected:Npe 692
 \cs_gset_protected:Npn 3178, 3194, 3209, 3223, 3246, 3258
 \cs_if_eq:NNTF 142, 3304
 \cs_if_exist:NTF 45, 95, 3154, 3190, 3205, 3219, 3243, 3255
 \cs_if_exist_use:NTF 1256
 \cs_new:Npn 112, 117,
201, 271, 273, 280, 286, 297, 311, 362, 526, 645, 1283, 1285
 \cs_new_eq:NN 267, 268, 269, 363, 728,
953, 962, 1023, 1187, 3176, 3245, 3257, 3316, 3376, 3461
 \cs_new_protected:Npe 3317
 \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, 862, 902, 916, 942, 949, 954, 963,
1010, 1012, 1018, 1049, 1066, 1078, 1088, 1131, 1143,
1148, 1154, 1174, 1180, 1188, 1203, 1227, 1261, 1268,
1278, 1287, 1292, 1336, 1362, 1391, 1411, 1483, 1585,
1591, 1623, 1634, 1651, 1678, 1709, 1737, 1747, 1766,
1781, 1797, 1806, 1825, 1834, 1837, 1861, 1906, 1924,
1936, 1974, 1995, 2023, 2134, 2144, 2150, 2168, 2211,
2225, 2240, 2250, 2281, 2325, 2345, 2351, 2429, 2451,
2519, 2533, 2561, 2591, 2597, 2603, 2609, 2638, 2694,
2700, 2706, 2714, 2720, 2730, 2747, 2788, 2831, 2842,
2854, 2867, 2880, 2891, 2907, 2909, 2947, 2975, 2994,
3012, 3028, 3073, 3102, 3119, 3150, 3167, 3173, 3188,
3203, 3217, 3239, 3293, 3302, 3400, 3448, 3454, 3456, 3462
 \cs_set:Npn 276, 435, 1230, 1236
 \cs_set_eq:NN 129,
134, 135, 136, 240, 3193, 3208, 3222, 3342, 3343, 3344,
3345, 3346, 3353, 3356, 3357, 3366, 3367, 3368, 3369, 3370
 \cs_set_protected:Npe 241, 243
 \cs_set_protected:Npn ... 3326, 3335, 3364, 3385, 3391
 \cs_to_str:N 2911, 2980, 3171, 3177, 3181
 \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 23, 41, 44
 \ctex_if_format_at_least:nTF 42
 \ctex_patch_cmd:Nnn 23
 \ctex_replace_package:nn 43

D

Default 369
 \defaultCJKfontfeatures 10
 depth 18

dim commands:

\dim_case:nn 1156
 \dim_case:nnTF 1106, 2173
 \dim_compare:nNnTF 890, 893, 1090, 1945
 \dim_compare_p:nNn 2380, 2386, 2390
 \dim_const:Nn 1265
 \dim_eval:n 203
 \dim_gset:Nn 1265
 \dim_if_exist:NTF 1264
 \dim_max:nn 2291
 \dim_new:N 52, 882, 1423, 1424, 1425, 1426
 \dim_set:Nn 887, 889,
2228, 2230, 2242, 2244, 2285, 2287, 2289, 2306, 2308, 2310
 \dim_set_eq:NN 2256, 2260
 \dim_use:N 224
 \dim_zero:N 2252, 2253
 \c_zero_dim
 894, 1951, 2048, 2234, 2268, 2296, 2381, 2387, 2391

E

else commands:

\else: 146, 256, 261, 309, 317, 1242, 1259
 EmboldenFactor 4
 \emph 1435, 1437, 1441, 1444, 1451, 1468, 1473, 3289
 \end 1460
 \enquote 1440
 \ensuremath 1516, 1558
 \env 1438
 EnvCS 4
 EnvCS+ 4
 EnvCS- 4

exp commands:

\exp_after:wN 140, 235, 253, 254, 259, 260, 262, 263, 274, 302
 \exp_args:Nc 344, 1280, 1758, 2127, 2140, 2355, 3176
 \exp_args:Ncc 1232
 \exp_args:NcNc 3169
 \exp_args:Ne ... 1501, 1561, 2029, 2577, 2618, 2772, 2911
 \exp_args:NNNo 219, 223, 226
 \exp_args:NNV 3082
 \exp_args:NnV 3130, 3135, 3140
 \exp_args:NNVV 3094
 \exp_args:NV 1498, 1507, 1511, 1515, 1530, 1557,
1884, 1900, 1958, 1989, 2016, 2337, 2447, 2514, 2551, 3075
 \exp_args:NVV 2507
 \exp_last_unbraced:Ne 114
 \exp_not:N 83, 242, 244, 258, 697
 \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 7
 experiment/halfright-prebreakpenalty 7
 experiment/punct-measure-fix 7

F

FallBack 11
 \fbox 3309

fi commands:

\fi: 146, 235, 264, 265, 309, 317, 1242, 1259

\file 1436, 1465

format 17

fp commands:

\fp_compare:nNnTF 372

\fp_eval:n 204

FullLeft 369

FullRight 369

G

gap 18

group commands:

\group_align_safe_begin: 246, 3405

\group_align_safe_end: 242, 244, 3455

\group_begin: 215, 275, 1229, 1478, 3169

\c_group_begin_token 722, 2147

\group_end: 226, 303, 1252, 1481, 3169

\c_group_end_token 728, 2347

H

HalfLeft 388

HalfRight 388

HangulJamo 388

HangulJamoL 388

HangulJamoT 388

HangulJamoV 388

hbox commands:

\hbox:n 80

\hbox_set:Nn 2403, 2437

\hbox_set:Nw 216

\hbox_set_end: 219, 223

\hbox_unpack:N 2405, 2439

\hbox_unpack_drop:N 2523

height 18

hidden 17

hyp internal commands:

_hyp_target_raise:n 80

I

if commands:

\if_catcode:w 257

\if_cs_exist:w 306

\if_dim:w 1255

\if_int_compare:w 232, 315, 1241

\if_meaning:w 251

\IfBooleanT 531

InlineEnv 4

InlineEnv+ 4

InlineEnv- 4

int commands:

\int_case:nNnTF 2415

\int_compare:nNnTF 150, 163, 183, 375, 956, 958, 1364,
1367, 1680, 1683, 1711, 1714, 1783, 1808, 1839, 1908, 2026

\int_const:Nn 69, 357, 380, 386, 1238

\int_div_truncate:nn 206

\int_eval:n 281, 288, 298, 312

\int_gdecr:N 1778, 1902, 1932, 2363, 2516

\int_gincr:N 1263, 1270, 1749, 2025

\int_gset:Nn 211, 1276, 2406

\int_gset_eq:NN 994, 1027, 2800

\int_if_exist:NnTF 341, 354

\int_incr:N 576, 586

\int_max:nn 566

\int_min:nn 565

\int_new:N 50, 209, 236, 237, 1275, 1312, 1313, 2367

\int_set:Nn 561, 565, 566, 571, 572, 582, 591

\int_set_eq:NN 562

\int_step_inline:nn 1370, 1717

\int_until_do:nNnn 27

\int_use:N ... 371, 1657, 1664, 1671, 1689, 1695, 1701,
1751, 1754, 1756, 1759, 1762, 1771, 1774, 1819, 1850,
1873, 1878, 1917, 2030, 2033, 2036, 2040, 2042, 2044,
2051, 2055, 2059, 2062, 2065, 2068, 2071, 2075, 2081,
2094, 2103, 2112, 2118, 2123, 2128, 2138, 2141, 2158,
2161, 2356, 2376, 2434, 2456, 2461, 2505, 2511, 2524,
2528, 2540, 2572, 2582, 2600, 2717, 2752, 2759, 2767, 2777

\int_value:w 207

\int_zero:N 725

\c_one_int 329, 375

\c_zero_int 66, 331, 956, 958,
1365, 1368, 1681, 1684, 1712, 1715, 1784, 1809, 1840, 1909

\item 1450, 1456

K

KaiMingPunct 5

KaiMingPunct+ 5

KaiMingPunct- 5

keys commands:

\keys_define:nn 321, 3322, 3331, 3379

\keys_set:nn 611, 613, 2957

L

\lastnodetype 2410

\LaTeX 1435, 1465, 3283

\LaTeXe 3235

LatinPunct 6

LoadFandol 9

LocalConfig 3

LongPunct 5

LongPunct+ 5

LongPunct- 5

M

\makexeCJKactive 324, 328

\makexeCJKinactive 325, 328

Mapping 9

\mbox 3308

MiddlePunct 5

MiddlePunct+ 5

MiddlePunct- 5

mode commands:

\mode_if_horizontal:TF 1393, 1912, 1938

\mode_if_math:TF 1787, 1812, 1843

\mode_leave_vertical: 1790, 1815, 1846

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 9

\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

para commands:

\para_end: 7

peek commands:

\peek_after:Nw 247, 253, 3463

\peek_meaning_remove:NtF 995, 1028

\l_peek_token 251, 258, 1001, 1034, 3408, 3411, 3416, 3422, 3425, 3428, 3434, 3437

\pkg 1437, 1442, 1451, 3275

PlainEquation 4

PoZheHao 388

PoZheHaoLigature 5

prg commands:

\prg_do_nothing: 129, 131, 268, 269, 651, 697, 953, 1187, 3316

\prg_new_conditional:Npnn 23, 143, 304, 313, 883, 1239, 1253, 1612, 2200, 2913

\prg_new_protected_conditional:Npnn 1491, 1525, 1541, 1573, 1601, 2370, 2397

\prg_return_false: 26, 146, 309, 317, 896, 898, 900, 1242, 1259, 1496, 1504, 1518, 1533, 1545, 1564, 1567, 1581, 1608, 1619, 2207, 2395, 2423, 2425, 2920

\prg_return_true: 26, 146, 309, 317, 895, 1242, 1259, 1503, 1509, 1513, 1517, 1532, 1548, 1551, 1559, 1563, 1577, 1580, 1609, 1615, 1618, 2203, 2206, 2394, 2422, 2916, 2919

prop commands:

\prop_gput:Nnn . 2846, 2858, 2871, 2884, 2895, 2908, 3094

\prop_if_in:NnTF 2844, 2856, 2869, 2882, 2893, 2915, 2918, 3082

\prop_map_inline:Nn 3105

\prop_new:N 1315, 1316, 1317

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, 1395, 1488, 1868, 2085, 3422, 3461

\s_stop 115, 117

sep 18

seq commands:

\seq_gpop:NN 1799, 1827, 1863, 1926

\seq_gpush:Nn 1785, 1788, 1791, 1810, 1813, 1816, 1841, 1844, 1847, 1910, 1914, 1920

\seq_gput_right:Nn 347, 348, 359, 708, 3114

\seq_gset_eq:NN 701

\seq_gset_from_clist:Nn 703

\seq_map_inline:Nn 731, 744, 757, 779, 809, 3152

\seq_new:N 337, 338, 700, 702, 705, 1314, 1318, 1326

\seq_pop_left:NN 3122, 3124

\seq_set_split:Nnn 3121

\setCJKfallbackfamilyfont 11

\setCJKfamilyfont 9

\setCJKmainfont 9

\setCJKmathfont 10

\setCJKmonofont 9

\setCJKsansfont 9

skip 17

skip commands:

\skip_const:Nn 1272

\skip_gset:Nn 1272

\skip_horizontal:N . 861, 951, 968, 975, 976, 983, 985, 1097, 1098, 1112, 1122, 1138, 1139, 1145, 1184, 1198, 1213, 1216, 1219, 1220, 1225, 1646, 2002, 2010, 2018, 2341

\skip_horizontal:n 2711

\skip_if_eq:nnTF 152, 161, 169, 186, 972, 1205, 2339, 2566, 2629

\skip_if_exist:NtF 1271

\skip_if_finite:NtF 885

\skip_new:N . 54, 989, 1419, 1420, 1421, 1422, 3330, 3377

`\skip_set:Nn` 220, 224,
 227, 888, 2227, 2232, 2266, 2298, 2315, 2333, 2624, 2626
`\skip_set_eq:NN`
 906, 965, 1068, 1192, 1638, 1997, 2154, 2335, 2565
`\skip_use:N`
 220, 227, 1674, 2052, 2056, 2060, 2097, 2106, 2159
`\c_zero_skip` 152, 161, 169, 186, 2340, 2630
`SlantFactor` 4
`\SplitArgument` 553
str commands:
`\c_backslash_str` 3079, 3086, 3110, 3157
`\str_case:nn` 1397, 1852, 3030, 3126
`\str_case:nnTF` 2996, 3015
`\str_const:Nn` 370
`\str_if_eq:nnTF` 17, 544, 733, 737,
 746, 759, 781, 811, 1889, 2283, 2476, 2521, 2611, 2616,
 2642, 2655, 2657, 2675, 2738, 2740, 2742, 2801, 2808, 2814
`subtract` 17
`symbol` 17
sys commands:
`\sys_if_engine_xetex:TF` 10

T

`\TeX` 3290
TeX and L^AT_EX 2_ε commands:
`\(` 21, 51
`\)` 21
`\@beginDvi` 24, 91, 95, 98
`\@empty` 24
`\@ifnextchar` 8
`\@ifpackagelater` 19
`\@ifframebox` 3257, 3258
`\@imakebox` 3245, 3246
`\@pkgextension` 28
`\[` 4
`\$` 84
`\addCJKfontfeatures` 10
`\AtBeginDocument` 8, 52
`\AtBeginDvi` 24
`\AtBeginShipout` 20, 24
`\baselineskip` 3
`\begin` 4, 19
`\begingroup` 38
`\box` 51
`\catcode` 20, 27
`\char` 21, 22
`\CJKecglue` 8, 42, 43, 45, 51
`\CJKfamily` 9, 20
`\CJKfamilydefault` 10, 11, 13
`\CJKfontspec` 10
`\CJKglue` 8, 20, 41, 51
`\CJKrmdefault` 10
`\CJKsout` 18, 19, 21
`\CJKsymbol` 20
`\CJKKunderanyline` 18, 19
`\CJKKunderanysymbol` 18, 19

`\CJKKunderdblline` 18, 19, 21
`\CJKKunderdot` 17–19, 21
`\CJKKunderline` 17–19, 21
`\CJKKunderwave` 17–19, 21
`\CJKxout` 19, 21
`\color` 17
`\color@b@x` 51
`\color@endgroup` 3298
`\color@setgroup` 3298
`\colorbox` 21
`\cprotect` 20, 21
`\defaultCJKfontfeatures` 10
`\document` 24
`\dotfill` 45
`\edef` 84
`\emph` 21
`\end` 4, 19
`\endgroup` 38
`\ensuremath` 21, 51
`\ExplSyntaxOff` 7
`\ExplSyntaxOn` 7
`\familydefault` 10
`\fbox` 21
`\fontdimen7` 42
`\footnote` 6, 20
`\footnotemark` 6
`\framebox` 21
`\global` 8
`\hbox` 19
`\hrulefill` 45
`\hskip` 3, 42, 43
`\Hy@SaveSpaceFactor` 80
`\hyper@anchorstart` 80
`\hyperref` 21, 22
`\icprotect` 21
`\kern` 7, 42, 46
`\lastskip` 42
`\leaders` 17, 43, 45
`\long` 8
`\makeatletter` 7
`\makeatother` 7
`\makebox` 21, 51
`\maxdimen` 14–16
`\mbox` 3, 8, 21, 51
`\meaning` 27
`\newCJKfontfamily` 5, 9
`\nfss@text` 2487
`\nobreak` 6
`\normalfont` 10
`\normalspacedchars` 14
`\null` 50, 51
`\outer` 20, 21
`\par` 4, 7
`\parfillskip` 7
`\path` 3
`\protected` 84

<code>\ref</code>	21, 22	tex commands:	
<code>\relax</code>	21, 83	<code>\tex_afterassignment:D</code>	3458
<code>\rmfamily</code>	9, 10	<code>\tex_font:D</code>	
<code>\savebox</code>	8		140, 145, 154, 155, 156, 166, 173, 174, 179, 180, 191
<code>\sbox</code>	8, 51	<code>\tex_fontdimen:D</code>	154, 155, 156, 166, 173, 174, 179, 180, 191
<code>\setbox</code>	8, 42, 50, 51	<code>\tex_global:D</code>	2146
<code>\setCJKfallbackfamilyfont</code>	11	<code>\tex_glueshrink:D</code>	197, 894, 2231, 2245
<code>\setCJKfamilyfont</code>	5, 9	<code>\tex_gluestretch:D</code>	196, 2229, 2243
<code>\setCJKmainfont</code>	2, 9	<code>\tex_hbox:D</code>	2147, 3295
<code>\setCJKmathfont</code>	10	<code>\tex_hskip:D</code>	3319, 3320
<code>\setCJKmonofont</code>	9	<code>\tex_iffontchar:D</code>	145
<code>\setCJKsansfont</code>	9	<code>\tex_italiccorrection:D</code>	995, 997, 1028, 1030
<code>\sffamily</code>	9, 10	<code>\tex_kern:D</code>	1289, 1290, 2048
<code>\shipout</code>	24	<code>\tex_lastkern:D</code>	1090, 1106, 1156, 1257
<code>\spacefactor</code>	26, 51, 80	<code>\tex_lastnodetype:D</code>	42, 958, 1241, 2407
<code>\textbf</code>	21	<code>\tex_lastskip:D</code>	46, 220,
<code>\textcolor</code>	3, 21		906, 965, 1068, 1192, 1258, 1638, 1997, 2097, 2154, 2565
<code>\textit</code>	21	<code>\tex_let:D</code>	3459
<code>\textnormal</code>	10	<code>\tex_nullfont:D</code>	142
<code>\textrm</code>	9, 10	<code>\tex_par:D</code>	7
<code>\textsf</code>	9, 10	<code>\tex_penalty:D</code>	66, 68, 1183
<code>\texttt</code>	9, 10	<code>\tex_romannumeral:D</code>	255
<code>\ttfamily</code>	9, 10	<code>\tex_setbox:D</code>	2146, 3295
<code>\UL@start</code>	43	<code>\tex_spacefactor:D</code>	994, 1027, 2800
<code>\UL@stop</code>	43	<code>\tex_spaceskip:D</code>	152, 158, 161, 184, 190, 196, 197
<code>\uline</code>	19	<code>\tex_the:D</code>	140, 647
<code>\ULon</code>	19	<code>\tex_unkern:D</code>	1295
<code>\ULthickness</code>	18	<code>\tex_unskip:D</code>	909, 966, 1071, 1193, 1298, 1641, 1998, 2163
<code>\unskip</code>	7	<code>\tex_XeTeXcharclass:D</code>	272, 375, 585, 593, 598
<code>\url</code>	8	<code>\tex_XeTeXdashbreakstate:D</code>	725
<code>\verb</code>	8	<code>\tex_XeTeXinterchartokenstate:D</code>	329, 331
<code>\verbatim@font</code>	8	<code>\tex_XeTeXinterchartoks:D</code>	641, 647, 695
<code>\vrule</code>	51	<code>\tex_XeTeXrevision:D</code>	371
<code>\xeCJK@document@hook</code>	70, 73	<code>\tex_XeTeXversion:D</code>	371
<code>\xeCJK@document@left@hook</code>	71, 75	<code>\tex_xspaceskip:D</code>	169, 177, 186, 194
<code>\xeCJK@document@right@hook</code>	72, 77	<code>\textbf</code>	1436, 1465, 1471
<code>\xeCJK@first@begindvi</code>	91, 92	textformat	17
<code>\xeCJKCancelSubCJKBlock</code>	13	<code>\texttt</code>	3237, 3270, 3272, 3291
<code>\xeCJKchar</code>	22	thickness	18
<code>\xeCJKDeclareCharClass</code>	13	tl commands:	
<code>\xeCJKDeclarePunctStyle</code>	5, 14	<code>\c_space_tl</code>	3376
<code>\xeCJKDeclareSubCJKBlock</code>	13	<code>\tl_clear:N</code>	2170, 2954, 2955
<code>\xeCJKEditPunctStyle</code>	14	<code>\tl_const:Nn</code>	28, 97, 148, 709, 1480
<code>\xeCJKfntefbox</code>	18	<code>\tl_gclear:N</code>	1300, 1891, 2032,
<code>\xeCJKfntefon</code>	19		2034, 2039, 2041, 2043, 2079, 2130, 2360, 2436, 2463, 2593
<code>\xeCJKknobreak</code>	6, 20	<code>\tl_gput_right:Nn</code>	80, 82, 84, 96
<code>\xeCJKOffVerbAddon</code>	20	<code>\tl_gset:Nn</code>	1281, 1372, 1375,
<code>\xeCJKResetCharClass</code>	6		1654, 1668, 1686, 1692, 1698, 1721, 1743, 1761, 1768,
<code>\xeCJKRestoreSubCJKBlock</code>	13		1818, 1849, 1916, 2050, 2054, 2058, 2061, 2064, 2067,
<code>\xeCJKsetkern</code>	14-16		2070, 2073, 2091, 2100, 2111, 2122, 2137, 2157, 2160, 2817
<code>\xeCJKsetup</code>	2, 3, 14, 15, 17-19	<code>\tl_gset_eq:NN</code>	1758, 2127, 2443
<code>\xeCJKsetwidth</code>	6, 14	<code>\tl_head:n</code>	1502, 1562
<code>\xeCJKShipoutHook</code>	20	<code>\tl_if_blank:nTF</code>	561
<code>\xeCJKVerbAddon</code>	8, 20	<code>\tl_if_blank_p:n</code>	558, 559
<code>\XeTeXdashbreakstate</code>	40	<code>\tl_if_empty:NTF</code>	668, 672, 681, 1373, 1495, 1719, 1893,
<code>\XeTeXinterchartoks</code>	21, 26		1898, 1940, 1977, 2000, 2115, 2490, 2499, 2512, 2958, 2963

\tl_if_empty:nTF	278, 284, 290, 298, 1544
\tl_if_eq:NNTF	1185, 1607
\tl_if_eq:NnTF	1377, 1381, 1661, 1723, 1727, 1773, 1802, 1830, 1866, 1929, 1986, 2006, 2087, 2109, 2117, 2441, 2546, 2613, 2644, 2647, 2659, 2662, 2677, 2680, 2708, 2722, 2749, 2756, 2764, 3034, 3042
\tl_if_exist:NNTF	25, 1340, 1750
\tl_if_head_eq_meaning:nNTF	1507, 1511, 1515, 1547, 1550, 1557
\tl_if_head_is_group:nTF	1498, 1553
\tl_if_head_is_N_type:nTF	121
\tl_if_novalue:nTF	565, 566
\tl_if_single:nTF	119
\tl_if_single_token:nTF	2977
\tl_map_inline:nn	597
\tl_new:N	48, 49, 87, 88, 89, 110, 1277, 1322, 1323, 1324, 1325, 1331, 1332, 1333, 1334, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1417, 1418, 1753, 1755, 2369, 2428
\tl_put_right:Nn	91, 109
\tl_replace_all:Nnn	683, 1487
\tl_reverse:N	1529
\tl_set:Nn	666, 670, 679, 1485, 1555, 1604, 1605, 1739, 1870, 1875, 1890, 2175, 2176, 2177, 2178, 2179, 2182, 2186, 2187, 2195, 2328, 2329, 2373, 2431, 2442, 2453, 2458, 2478, 2479, 2494, 2495, 2502, 2535, 2733, 2737, 2739, 2741, 2743, 2771, 2782, 2784, 2956, 2980, 2983, 2989, 2998, 2999, 3000, 3001, 3002, 3007, 3017, 3018, 3019, 3058, 3068, 3089
\tl_set_eq:NN	1899, 2513
\tl_tail:n	1556
\tl_to_str:n	3006, 3023
\tl_trim_left_spaces:n	1556
\tl_trim_right_spaces:n	1606
\tl_trim_spaces:n	115, 122, 125, 1486
\tl_use:N	74, 76, 78, 105, 1740, 1770, 1872, 1877, 2455, 2460, 2504, 2537, 2569, 2579, 2620, 2625, 2627, 2697, 2703, 2726, 2774
\tn	1429, 1436, 1472, 1473, 1474, 1537, 1538, 3162, 3166, 3232, 3233, 3234, 3236, 3238, 3269, 3270, 3271, 3272, 3273, 3274, 3276, 3277, 3279, 3280, 3282, 3283, 3286, 3288, 3290
token commands:	
\c_math_toggle_token	3402
\c_space_token	251
\token_if_cs:NNTF	2979
\token_if_eq_meaning:NNTF	3410, 3422, 3436
\token_if_group_begin:NNTF	3416, 3425
\token_if_group_end:NNTF	3428
\token_if_letter:NNTF	569
\token_if_macro:NNTF	3408, 3434
\token_if_space:NNTF	1001, 1034
\token_to_meaning:N	274, 302
\TrimSpaces	528
U	
use commands:	
\use:N	527, 710, 1258, 1284, 1286, 2710, 3181
\use_i:nn	865
\use_none:n	231, 234, 235
\use_none:nn	177, 194
V	
Verb	8
W	
WidowPenalty	4
X	
xCJKecglue	3, 3331
xeCJK commands:	
\xeCJK_add_to_shipout:n	108, 108
\xeCJK_allow_break:	65, 65
\xeCJK_app_inter_class_toks:nnn ..	658, 658, 663, 840
\xeCJK_Boundary_and_Default:	839, 843, 843
\xeCJK_Boundary_and_NormalSp:	1009, 1010, 1010
\xeCJK_check_for_ecglue: ..	846, 962, 3345, 3356, 3369
\xeCJK_check_for_ecglue_normalsp:	1011, 1023, 3347, 3358, 3371
\xeCJK_check_for_glue:	1042, 1049, 1049
\xeCJK_check_for_xglue: ..	1178, 1187, 3344, 3355, 3368
\xeCJK_CJK_and_Boundary:w	3399, 3400, 3400
\xeCJK_class_group_begin:	720, 720, 823, 1043
\xeCJK_class_group_end:	720, 728, 832, 3451, 3452
\xeCJK_class_num:n	526, 526, 547, 550, 591, 598, 641, 642, 647, 648, 696, 697
\xeCJK_clear_Boundary_and_CJK_toks:	688, 688, 692, 694, 826, 1044
\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, 3354, 3355
\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	1268, 1307
\xeCJK_declare_node:n	1261, 1261, 1302, 1303, 1304, 1305, 1306, 1308, 1309, 1310
\xeCJK_fallback_symbol:NN	795, 827, 1046
\xeCJK_font_gset_to_current:N	139, 139
\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, 2047, 2053, 2057, 3327, 3336, 3365
\xeCJK_glyph_if_exist:N	143
\xeCJK_glyph_if_exist:NNTF	143
\xeCJK_glyph_if_exist_p:N	143
\xeCJK_if_CJK_class:N	304
\xeCJK_if_CJK_class:NNTF	304

- \xeCJK_if_CJK_class_p:N 304
- \xeCJK_if_last_node:n 1253
- \xeCJK_if_last_node:nTF 864, 866,
869, 872, 918, 921, 924, 927, 967, 970, 1020, 1194, 1197,
1209, 1212, 1215, 1253, 1617, 2192, 2202, 2205, 2213, 2327
- \xeCJK_if_last_node:TF 1956, 2171, 2254, 2258
- \xeCJK_if_last_node_p:n 1253
- \xeCJK_if_last_punct:TF 1176
- \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_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, 992, 1008, 1025, 1039, 3399
- \xeCJK_make_boundary: 319, 319
- \xeCJK_make_node:n
..... 998, 1003, 1031, 1036, 1261, 1278, 2804, 2810, 2824
- \xeCJK_make_space_node:
..... 1002, 1035, 3316, 3316, 3343, 3354, 3367
- \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
- \xeCJK_peek_catcode_ignore_spaces:N 238, 238, 3402
- \xeCJK_pre_inter_class_toks:nnn
..... 652, 652, 657, 789, 790, 1006
- \xeCJK_remove_node:
..... 867, 870, 873, 929, 944, 968, 975, 1021,
1058, 1094, 1110, 1116, 1119, 1159, 1161, 1163, 1165,
1167, 1195, 1198, 1210, 1213, 1216, 1292, 2194, 2215, 2218
- \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, 1045
- \xeCJK_set_char_class:nnn 546, 550, 579, 579
- \xeCJK_space_glue: 1406, 2048, 3353, 3376, 3386
- \xeCJK_space_or_xecglue:
..... 1021, 1402, 3342, 3353, 3366, 3451
- \c_xeCJK_space_skip_tl 148, 888, 974, 1207, 2334
- \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: 1120, 1163, 1404, 2658
- 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 1185
- \c_xeCJK_alignment_tl 1480, 1488
- __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
..... 1314, 1785, 1788, 1791, 1800, 1810, 1813,
1816, 1828, 1841, 1844, 1847, 1864, 1910, 1914, 1920, 1927
- l_xeCJK_boundary_active_tl
... 1334, 1801, 1802, 1829, 1830, 1865, 1866, 1928, 1929
- __xeCJK_boundary_box_begin: 1312, 1792, 2134
- __xeCJK_boundary_box_end:n 1312, 1803, 2345
- __xeCJK_boundary_box_end_transparent:n
..... 1312, 2500, 2519
- __xeCJK_boundary_box_finish:nn 1312, 2348, 2357, 2451
- __xeCJK_boundary_box_open:N 1312, 2140, 2144
- __xeCJK_boundary_box_set_last_from_node: ...
..... 1312, 2429, 2471
- __xeCJK_boundary_capture_allocate:n
..... 1312, 1336, 1358, 1359, 1360, 2029
- __xeCJK_boundary_capture_begin:
..... 51, 1312, 1817, 1848, 1915, 2023, 2136
- __xeCJK_boundary_capture_class:n
..... 822, 833, 845, 1041, 1312, 1362, 1413, 1854, 1855
- g_xeCJK_boundary_capture_depth_int
..... 956, 1312, 1365,
1370, 1657, 1664, 1671, 1681, 1689, 1695, 1701, 1712,
1717, 1819, 1850, 1873, 1878, 1902, 1917, 1932, 2025,
2027, 2030, 2033, 2036, 2040, 2042, 2044, 2051, 2055,
2059, 2062, 2065, 2068, 2071, 2075, 2081, 2094, 2103,
2112, 2118, 2123, 2128, 2138, 2141, 2158, 2161, 2356,
2363, 2376, 2434, 2456, 2461, 2505, 2511, 2516, 2524,
2528, 2540, 2572, 2582, 2600, 2717, 2752, 2759, 2767, 2777
- __xeCJK_boundary_capture_emit_left:nn
..... 1312, 1379, 1383, 1725, 1729, 1737
- __xeCJK_boundary_capture_report_first:n
..... 1312, 1709, 2496
- __xeCJK_boundary_capture_resume: . 1312, 1766, 3299
- __xeCJK_boundary_capture_space: . 1312, 2084, 2150
- __xeCJK_boundary_capture_suspend: . 1312, 1747, 3297
- __xeCJK_boundary_CJK_and_math: 3454
- __xeCJK_boundary_emit_left:nn 1312, 2507, 2597
- __xeCJK_boundary_emit_left:nnn
..... 1312, 1741, 2599, 2603, 2692
- __xeCJK_boundary_emit_left_math_space:nnn ...
..... 1312, 2606, 2609
- __xeCJK_boundary_emit_left_normal:nnn
..... 1312, 2607, 2638
- __xeCJK_boundary_enable_source_space:n
..... 1312, 2805, 2811, 2825, 2831
- l_xeCJK_boundary_first_tl 1332,
1870, 1893, 1899, 2453, 2478, 2490, 2494, 2499, 2509, 2513
- __xeCJK_boundary_group_end:n . 3418, 3431, 3439, 3441
- __xeCJK_boundary_group_math:w 3417, 3426, 3456
- __xeCJK_boundary_group_math_brace: 3459, 3461
- __xeCJK_boundary_group_math_branches: 3463
- __xeCJK_boundary_group_math_peek: 3458, 3462

```

\_xeCJK_boundary_hmode_transparent_begin: ...
    ..... 1312, 1906, 2874, 3196, 3211, 3225
\_xeCJK_boundary_hmode_transparent_end: ...
    ..... 1312, 1924, 2876, 3198, 3213, 3227
\_xeCJK_boundary_identity:n ..... 3411, 3437
\_xeCJK_boundary_if_capture_box_sized: 1312, 2370
\_xeCJK_boundary_if_capture_box_sized:TF .. 2400
\_xeCJK_boundary_if_capture_box_visible: ...
    ..... 1312, 2397
\_xeCJK_boundary_if_capture_box_visible:TF . 2492
\_xeCJK_boundary_if_last_math_node: ... 1312, 1612
\_xeCJK_boundary_if_last_math_node:TF . 1625, 1642
\_xeCJK_boundary_if_last_math_space: .. 1312, 2200
\_xeCJK_boundary_if_last_math_space:TF .....
    ..... 875, 934, 979, 1080
\_xeCJK_boundary_if_math_edge:n ..... 1312, 1573
\_xeCJK_boundary_if_math_head:n ..... 1312, 1491
\_xeCJK_boundary_if_math_head:nTF . 1501, 1576, 1587
\_xeCJK_boundary_if_math_tail:n ..... 1312, 1525
\_xeCJK_boundary_if_math_tail:nTF . 1561, 1579, 1593
\_xeCJK_boundary_if_math_tail_reversed:n .. 1542
\_xeCJK_boundary_if_math_tail_reversed:nTF . 1530
\_xeCJK_boundary_if_math_tail_space:n . 1312, 1601
\_xeCJK_boundary_if_math_tail_space:nTF ... 1595
\_xeCJK_boundary_if_registered:n ..... 1312, 2913
\_xeCJK_boundary_if_registered:nTF .... 3075, 3107
\_xeCJK_boundary_inline_box_begin: 1312, 1781, 2861
\_xeCJK_boundary_inline_box_end:n . 1312, 1797, 2863
\_xeCJK_boundary_inline_last_box_begin: ....
    ..... 1312, 1806, 2848, 3248, 3260
\_xeCJK_boundary_inline_last_box_end:n .....
    ..... 1312, 1825, 2850, 3252, 3264
\_xeCJK_boundary_inline_stream_begin:n .....
    ..... 1312, 1834, 2897
\_xeCJK_boundary_inline_stream_begin:nn ....
    ..... 1312, 1835, 1837
\_xeCJK_boundary_inline_stream_end:n .....
    ..... 1312, 1861, 2899
\_xeCJK_boundary_last_box_end:n .. 1312, 1831, 2351
\l_xeCJK_boundary_last_tl ..... 1333,
    1875, 1890, 1898, 1899, 1900, 2458, 2479, 2495, 2512,
    2513, 2514, 2733, 2737, 2739, 2741, 2743, 2771, 2782, 2784
\l_xeCJK_boundary_math_actual_skip .....
    1420, 2106, 2227, 2229, 2231, 2232, 2243, 2245, 2266, 2320
\_xeCJK_boundary_math_begin: ..... 1312, 1391, 1414
\_xeCJK_boundary_math_begin:n 1585, 3180, 3249, 3261
\_xeCJK_boundary_math_begin_visible: .....
    ..... 1312, 1411, 1588
\l_xeCJK_boundary_math_delta_skip .....
    ..... 1422, 2298, 2315, 2340, 2341, 2630, 2633
\l_xeCJK_boundary_math_ecglue_dim .....
    ..... 1424, 2287, 2293, 2301, 2308, 2312, 2318
\l_xeCJK_boundary_math_ecglue_skip .....
    ..... 1421, 2288, 2300, 2309, 2317, 2336, 2626
\_xeCJK_boundary_math_end:n .....
    ..... 1312, 1591, 3182, 3251, 3263
\_xeCJK_boundary_math_end_confirm: .....
    ..... 1312, 1626, 1653, 1678
\_xeCJK_boundary_math_end_confirm_node: ....
    ..... 1312, 1597, 1623
\_xeCJK_boundary_math_end_confirm_space: ...
    ..... 1312, 1596, 1634
\_xeCJK_boundary_math_end_confirm_space_-
    visible: ..... 1312, 1644, 1651
\_xeCJK_boundary_math_set:n ..... 1483, 1494, 1528
\l_xeCJK_boundary_math_shrink_dim .....
    ..... 1426, 2230, 2236, 2244, 2247, 2253, 2257, 2270
\l_xeCJK_boundary_math_skip .....
    ..... 1419, 1638, 1639, 1646, 1674, 2286, 2307, 2333, 2624
\_xeCJK_boundary_math_space_consume: .....
    ..... 876, 935, 982, 1312, 2211, 2330
\_xeCJK_boundary_math_space_decode: .....
    ..... 1312, 2183, 2216, 2250
\l_xeCJK_boundary_math_space_dim .....
    ..... 1423, 2285, 2294, 2306, 2313
\_xeCJK_boundary_math_space_encode: 1312, 2240, 2803
\_xeCJK_boundary_math_space_expire: .....
    ..... 1312, 1987, 2012, 2591
\_xeCJK_boundary_math_space_set_actual:n ...
    ..... 1312, 2225, 2577, 2618, 2772
\_xeCJK_boundary_math_space_set_delta:n ....
    ..... 1312, 2281, 2337, 2628
\_xeCJK_boundary_math_space_to_CJK: .....
    ..... 1083, 1169, 1171, 1312, 2325
\l_xeCJK_boundary_math_stretch_dim .....
    ..... 1425, 2228, 2235, 2242, 2246, 2252, 2261, 2269
\l_xeCJK_boundary_math_tl .....
    ..... 1417, 1485, 1487, 1495, 1499,
    1502, 1508, 1512, 1516, 1529, 1531, 1555, 1558, 1604, 1607
\l_xeCJK_boundary_math_type_tl 1418, 2328, 2329, 2338
\l_xeCJK_boundary_node_tl 1331, 1396, 1397, 1605,
    1607, 1739, 1742, 1869, 1885, 1976, 1977, 1986, 1990,
    1999, 2000, 2006, 2017, 2086, 2087, 2109, 2115, 2129,
    2440, 2441, 2442, 2444, 2502, 2508, 2535, 2544, 2546, 2551
\_xeCJK_boundary_peek_math:w ..... 3412, 3438
\_xeCJK_boundary_pop_node:N .....
    ..... 1312, 1396, 1869, 1976, 1999, 2086, 2168, 2440
\l_xeCJK_boundary_post_box ..... 1330,
    1944, 1947, 1948, 1949, 1953, 1981, 1985, 2003, 2011, 2015
\_xeCJK_boundary_post_transparent: 1312, 1936, 2887
\_xeCJK_boundary_post_transparent_relocate: .
    ..... 1312, 1952, 1974
\_xeCJK_boundary_post_transparent_relocate_-
    glue: ..... 1312, 1980, 1995
\_xeCJK_boundary_prepare_onearg_box:N .....
    ..... 1312, 3167, 3308, 3309
\_xeCJK_boundary_prepare_onearg_box:NN 1312, 3173
\_xeCJK_boundary_prepare_sbox: ..... 3302, 3315
\l_xeCJK_boundary_probe_box .....
    ..... 2368, 2403, 2409, 2437, 2446
\g_xeCJK_boundary_probe_last_tl .....
    ..... 2428, 2436, 2444, 2448

```



```

\l__xeCJK_boundary_probe_name_tl .....
..... 2369, 2373, 2381, 2387, 2391, 2405, 2431, 2439
\g__xeCJK_boundary_probe_type_int . 2367, 2406, 2415
\__xeCJK_boundary_register_box:nn .....
..... 1312, 2842, 3130, 3310, 3311
\__xeCJK_boundary_register_makeboxes: .....
..... 1312, 3239, 3312
\__xeCJK_boundary_register_post_transparent:n
..... 1312, 2880, 3146, 3313
\__xeCJK_boundary_register_stream:nn 1312, 2891, 3140
\__xeCJK_boundary_register_transparent:n ....
..... 1312, 2867, 3144
\__xeCJK_boundary_register_wrapped_box:nn ...
..... 1312, 2854, 3135
\g__xeCJK_boundary_registered_prop .. 1315, 2844,
2846, 2856, 2859, 2869, 2872, 2882, 2885, 2893, 2895, 2915
\__xeCJK_boundary_replay_before: .....
..... 1312, 1895, 1931, 2362, 2530, 2533
\__xeCJK_boundary_replay_before_math_space: ..
..... 1312, 2547, 2561
\__xeCJK_boundary_replay_node:n .....
1312, 1900, 1959, 1990, 2017, 2514, 2548, 2551, 2586, 2788
\__xeCJK_boundary_reserve:N .....
..... 1312, 2909, 3175, 3192, 3207, 3221
\__xeCJK_boundary_reserve:n .....
..... 1312, 2907, 2911, 3242, 3314
\__xeCJK_boundary_reserve_space: ..... 3413
\g__xeCJK_boundary_reserved_prop .. 1316, 2908, 2918
\__xeCJK_boundary_restore_space: .....
..... 1312, 2121, 2552, 2714
\__xeCJK_boundary_restore_space:n .....
..... 1312, 2650, 2664, 2683, 2687, 2689, 2716, 2720
\__xeCJK_boundary_sbox:Nn ..... 3293, 3304, 3305
\__xeCJK_boundary_set_last_from_node:n .....
..... 1312, 1885, 2448, 2730
\__xeCJK_boundary_set_math_tail_from_capture:
..... 1312, 1886, 2472, 2747
\g__xeCJK_boundary_suspend_depth_int .....
..... 1313, 1368, 1684, 1715, 1749, 1751, 1754,
1756, 1759, 1762, 1771, 1774, 1778, 1784, 1809, 1840, 1909
\__xeCJK_boundary_use_ccglue:n 1312, 2665, 2667, 2700
\__xeCJK_boundary_use_ecglue:n .....
..... 1312, 2649, 2652, 2669, 2682, 2685, 2694
\__xeCJK_boundary_use_glue:nn .....
..... 1312, 2632, 2696, 2702, 2706, 2725
\__xeCJK_boundary_use_ulem_glue:n ..... 43
\g__xeCJK_boundary_user_applied_seq 1318, 3114, 3152
\l__xeCJK_boundary_user_invalid_bool .....
..... 1320, 2952, 2961, 2966,
2968, 2970, 2984, 2990, 3008, 3024, 3037, 3046, 3056, 3066
\l__xeCJK_boundary_user_mode_given_bool .....
..... 1321, 2953, 3014, 3052, 3062
\l__xeCJK_boundary_user_mode_tl .....
..... 1324, 2956, 3017, 3018, 3019, 3034,
3042, 3055, 3058, 3065, 3068, 3092, 3125, 3131, 3136, 3141
\l__xeCJK_boundary_user_name_tl ..... 1322,
2954, 2958, 2980, 2983, 2989, 3076, 3079, 3083, 3086, 3096
\__xeCJK_boundary_user_register:n ..... 1312, 2947
\__xeCJK_boundary_user_register_apply: . 1312, 3102
\__xeCJK_boundary_user_register_apply:nn ....
..... 1312, 3113, 3119
\__xeCJK_boundary_user_register_check: . 1312, 3150
\g__xeCJK_boundary_user_register_closed_bool .
..... 1319, 2949, 3104
\__xeCJK_boundary_user_register_command:n ...
..... 1312, 2975
\__xeCJK_boundary_user_register_mode:n . 1312, 3012
\__xeCJK_boundary_user_register_store: .....
..... 1312, 2971, 3073
\__xeCJK_boundary_user_register_strategy:n ...
..... 1312, 2994
\__xeCJK_boundary_user_register_validate: ...
..... 1312, 2969, 3028
\g__xeCJK_boundary_user_registered_prop .....
..... 1317, 3083, 3095, 3105
\l__xeCJK_boundary_user_strategy_tl .....
..... 1323, 2955, 2963,
2998, 2999, 3000, 3001, 3002, 3007, 3030, 3091, 3123, 3126
\l__xeCJK_boundary_user_value_seq .....
..... 1326, 3121, 3123, 3125
\l__xeCJK_boundary_user_value_tl .. 1325, 3089, 3097
\__xeCJK_boundary_wrap_transparent_noarg:NN ..
..... 1312, 3188
\__xeCJK_boundary_wrap_transparent_onearg:NN .
..... 1312, 3203
\__xeCJK_boundary_wrap_transparent_onearg-
braced:NN ..... 1312, 3217
\__xeCJK_ccglue_or_space: .....
..... 1116, 1161, 1195, 1210, 1227, 3385, 3391
\l__xeCJK_ccglue_skip ..... 3327, 3330
\__xeCJK_check_for_ecglue: ... 843, 852, 854, 962, 3356
\__xeCJK_check_for_ecglue_aux: .... 42, 843, 858, 862
\__xeCJK_check_for_ecglue_normalsp: .....
..... 1016, 1018, 1024, 3359
\__xeCJK_check_for_glue_auxi: ..... 1052, 1154
\__xeCJK_check_for_glue_auxii: 1061, 1146, 1152, 1174
\__xeCJK_check_for_glue_auxiii: ..... 1177, 1180
\__xeCJK_check_for_glue_skip: ..... 1055, 1066
\__xeCJK_check_for_glue_skip_consume_and-
fallback: ..... 1076, 1148
\__xeCJK_check_for_glue_skip_kern: ..... 1073, 1078
\__xeCJK_check_for_glue_skip_kern_normal: ...
..... 1086, 1088
\__xeCJK_check_for_glue_skip_nonkern: .. 1074, 1131
\__xeCJK_check_for_glue_skip_restore: .....
..... 1084, 1100, 1126, 1128, 1141, 1143
\__xeCJK_check_for_xecglue: ..... 848, 3345, 3369
\__xeCJK_check_for_xecglue_normalsp: 1012, 3348, 3372
\__xeCJK_check_for_xglue: ..... 1188, 3344, 3368
\__xeCJK_check_for_xglue_aux: ..... 1199, 1203
\__xeCJK_check_num_range:nnNN ..... 555, 555, 581

```

\c__xeCJK_CJ_chars_clist	444, 634	_xeCJK_info:nnn	56, 64
\l__xeCJK_CJ_strict_bool	633, 637	_xeCJK_int_until_down	229, 231, 234, 235
_xeCJK_CJK_and_Boundary_relax:N	3423	\l__xeCJK_interword_dim	882, 889, 891
_xeCJK_CJK_and_math:	3403, 3448, 3455	\c__xeCJK_IS_chars_clist	443, 459
\c__xeCJK_CJK_chars_clist	463, 627	\c__xeCJK_iteration_marks_chars_clist	429
\g__xeCJK_CJK_class_seq	700	\l__xeCJK_last_kern_dim	2173, 2189, 2257, 2261
_xeCJK_CJK_class_tl:n	307, 311, 710	\g__xeCJK_last_node_tl	46, 1277, 1281, 1300, 1760, 1768, 1891, 1940, 1959, 2130, 2360, 2463, 2593, 2817
\l__xeCJK_CJK_group_bool	320, 723, 727	\l__xeCJK_last_penalty_bool	1182
\g__xeCJK_CJK_range_clist	619	\l__xeCJK_last_penalty_int	1183
\g__xeCJK_CJStarter_range_clist	626	\g__xeCJK_last_punct_tl	801, 803
\c__xeCJK_CL_chars_clist	419, 456	\l__xeCJK_last_skip	906, 907, 951, 965, 973, 976, 981, 983, 985, 989, 1068, 1069, 1098, 1112, 1122, 1139, 1145, 1184, 1192, 1206, 1220, 1225, 1997, 2002, 2010, 2018, 2154, 2155, 2159, 2565, 2567
\c__xeCJK_class_begin_int	380, 386	\l__xeCJK_latin_punct_bool	612, 616
_xeCJK_class_csname:n	341, 345, 349, 354, 357, 362, 362, 363, 527, 710	\c__xeCJK_left_tl	1185
\g__xeCJK_class_seq	337, 347, 359, 701, 731, 744, 757, 779, 809	_xeCJK_make_node:N	1280, 1287, 2189, 2246, 2247
\c__xeCJK_CM_chars_clist	513, 629	_xeCJK_make_space_node:	2816, 3317, 3343, 3367
\g__xeCJK_CM_range_clist	621	_xeCJK_msg_new:nn	56, 56, 364, 2924, 2929, 2931, 2933, 2935, 2937, 2939, 2944
\l__xeCJK_ecglue_skip	861, 968, 975, 1097, 1138, 1198, 1213, 1216, 1219, 2336, 3336, 3365, 3377	_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, 2950, 2982, 2988	\c__xeCJK_nobreak_penalty_int	68, 69
_xeCJK_error:nn	56, 59, 342, 355, 2960, 2965, 3005, 3022, 3036, 3044, 3054, 3064, 3078, 3085, 3109, 3156	_xeCJK_node:n	1090, 1108, 1115, 1117, 1158, 1160, 1162, 1164, 1166, 1168, 1170, 1283, 2175, 2176, 2177, 2178, 2179, 2180, 2185, 2187
\c__xeCJK_EX_chars_clist	441, 458	\g__xeCJK_node_int ..	1263, 1266, 1270, 1273, 1275, 1276
_xeCJK_font_gset_to_current_aux:NN	140, 141	\g__xeCJK_non_CJK_class_seq	700
\c__xeCJK_FullLeft_chars_clist	414, 608	\c__xeCJK_NormalSpace_chars_clist	400, 628
\g__xeCJK_FullLeft_range_clist	604	\g__xeCJK_NormalSpace_range_clist	620
\c__xeCJK_FullRight_chars_clist	454, 609	\c__xeCJK_NS_chars_clist	427, 457
\g__xeCJK_FullRight_range_clist	605	\c__xeCJK_OP_chars_clist	405, 416
_xeCJK_get_charcode:w	274, 280, 286, 297	_xeCJK_orig_iframebox:w	3255, 3257, 3262
_xeCJK_glue_check_expire_stale: 843, 877, 954, 1060		_xeCJK_orig_imakebox:w	3243, 3245, 3250
\g__xeCJK_glue_check_pending_bool	42, 46, 904, 931, 946, 947, 959, 1082, 1092, 1095, 1103, 1105, 1133, 1134, 1150, 1151, 1311, 1763, 1776, 1777, 1892, 2131, 2219, 2361, 2464, 2594, 2839, 3430	\c__xeCJK_package_ext_tl	25, 28
_xeCJK_glue_node:n	1285, 3319, 3320	_xeCJK_package_hook:nn	70, 85
\c__xeCJK_HalfLeft_chars_clist	400, 606	_xeCJK_peek_catcode_false:w	243, 263, 269
\g__xeCJK_HalfLeft_range_clist	602	_xeCJK_peek_catcode_ignore_spaces_branches:w	247, 249, 254
\c__xeCJK_HalfRight_chars_clist	400, 607	_xeCJK_peek_catcode_true:w	241, 260, 268
\g__xeCJK_HalfRight_range_clist	603	\l__xeCJK_peek_ignore_spaces_bool	27, 245, 252, 270, 3406, 3450
\g__xeCJK_HangulJamo_range_clist	622	\l__xeCJK_peek_search_token	240, 258, 267
\c__xeCJK_HangulJamoL_chars_clist	520, 630	\c__xeCJK_PO_chars_clist	453, 460
\g__xeCJK_HangulJamoL_range_clist	623	\l__xeCJK_pozhehao_ligature_bool	610, 615
\c__xeCJK_HangulJamoT_chars_clist	520, 632	\c__xeCJK_PR_chars_clist	412, 417
\g__xeCJK_HangulJamoT_range_clist	625	_xeCJK_punct_glue:NN	803
\c__xeCJK_HangulJamoV_chars_clist	520, 631	_xeCJK_punct_rule:NN	801
\g__xeCJK_HangulJamoV_range_clist	624	_xeCJK_recover_ecglue_source_space: 843, 857, 902	
\c__xeCJK_hyphens_chars_clist	427, 438, 461	_xeCJK_recover_ecglue_source_space_aux: ...	843, 910, 916
_xeCJK_if_last_glue:TF	46, 217, 850, 856, 1014, 1054, 1190, 1229, 1297, 1636, 1979, 2089, 2152, 2563	_xeCJK_recover_ecglue_source_space_fallback:	843, 912, 914, 953
_xeCJK_if_last_hlist:TF	1229, 1942, 2353	_xeCJK_recover_ecglue_source_space_restore:	843, 930, 936, 949
_xeCJK_if_last_kern:TF	1051, 1072, 1229, 1294		
_xeCJK_if_last_math:TF	1057, 1135, 1218, 1229, 1614		
_xeCJK_if_last_none:TF	1229		
_xeCJK_if_last_penalty:TF	1229		

_xeCJK_recover_ecglue_source_space_success:	
.....	843 , 919 , 922 , 925 , 942
_xeCJK_replace_space:	 851 , 963 , 963 , 1015
\l_xeCJK_reserve_space_bool	
.....	1111 , 1121 , 2066 , 3384 , 3390 , 3398
\c_xeCJK_right_tl	 801 , 803
_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 , 883
_xeCJK_skip_if_interword:NTF	
.....	907 , 981 , 1069 , 1639 , 2155
\g_xeCJK_space_factor_int	
.....	150 , 163 , 183 , 204 , 207 , 209 , 994 , 1027 , 2800
_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	
.....	1230 , 1245 , 1246 , 1247 , 1248 , 1249 , 1250 , 1251
_xeCJK_tmp:w	 276 , 302 , 435 , 438 , 439
_xeCJK_tmp_aux:NNn	 1232 , 1236
\l_xeCJK_tmp_bool	 53
\l_xeCJK_tmp_box	 51 , 216 , 224
\l_xeCJK_tmp_clist	 55 , 535 , 536
\l_xeCJK_tmp_dim	. 52 , 887 , 891 , 2289 , 2302 , 2310 , 2319
\l_xeCJK_tmp_int	 50 , 363 , 582 , 585 , 591 , 593
\l_xeCJK_tmp_skip	
.....	54 , 888 , 889 , 2049 , 2052 , 2053 , 2056 , 2057 , 2060
\l_xeCJK_tmp_tl	
.....	48 , 666 , 668 , 670 , 672 , 675 , 679 , 681 , 683 , 685
\l_xeCJK_tmpb_tl	 49
_xeCJK_ulem_glue:n	 43
_xeCJK_update_clear_toks:n	 690 , 712
_xeCJK_use_ecglue_skip:	
.....	860 , 867 , 870 , 945 , 1058 , 1165 , 1167
_xeCJK_warning:n	 60
_xeCJK_warning:nn	 56 , 61
_xeCJK_warning:nnn	 62
_xeCJK_warning:nnnn	 63
\l_xeCJK_xecglue_bool	
.....	1096 , 1137 , 2069 , 2331 , 3341 , 3352 , 3363 , 3378
\c_xeCJK_xetex_version_str	 370 , 372
xeCJKactive	 3 , 321
\xeCJKCancelSubCJKBBlock	 13
\xeCJKDeclareCharClass	 13 , 528
\xeCJKDeclarePunctStyle	 14
\xeCJKDeclareSubCJKBBlock	 13
\xeCJKEditPunctStyle	 14
\xeCJKfntefon	 19
\xeCJKnobreak	 20
\xeCJKOffVerbAddon	 20
\xeCJKResetCharClass	 14 , 617 , 638
\xeCJKResetPunctClass	 14 , 531 , 600 , 635
\xeCJKRestoreSubCJKBBlock	 13
\xeCJKsetkern	 14
\xeCJKsetup	 2
\xeCJKsetwidth	 14
\xeCJKShipoutHook	 20 , 90
\xeCJKVerbAddon	 20