

xeCJK 宏包

CTEX.ORG

2026/08/22 v3.10.6*

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	234
3.5 标点符号的处理	14	代码索引	241
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. 7023032e](#).

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 `\xeCJKsetup {key = val}` 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 {key_1 = val_1, key_2 = val_2, ...}
```

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

例 2

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

等价于

例 3

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

带有 **★** 或者 **☆** 标记的选项或命令只能在导言区中使用, 其中 **☆** 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 L^AT_EX 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 L^AT_EX 环境起止点视为可断行点。
InlineEnv- 如果有某些特殊的 L^AT_EX 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+ 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 *<family>* 的备用字体。例如, 使用

例 10

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

可以将 SimSun-ExtB 作为 SimSun 的备用字体。

```
FallBack FallBack = { [<font features>] {<font name>} }
```

xeCJK 在 ** 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>`: 以上选项计算的标点间空白宽度, 可能小于 `min-bound-to-kerning` 选项的结果。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 $\text{\XeTeX}$ 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> }
\underline \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} }
\underline \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:

瞻彼阒者, 虚室生白, 吉祥止止

```

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

```

thickness 设置 \CJKunderline、\CJKunderdblline 和 \CJKsout 的线的厚度。初始值是 \ULthickness。

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

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

sep 设置 \CJKunderdot 与 \CJKunderline、\CJKunderdblline 或 \CJKunderwave 嵌套使用时, 点与线或者线与点的距离。

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

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

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

```

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

```

瞻彼阒者, 虚室生白, 吉祥止止

此外还有允许自定义下划线和符号的 \CJKunderanyline 和 \CJKunderanysymbol。

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

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

例如, 高亮效果也可以如下实现:

```

1 \CJKunderanyline*{0.5ex}{\color{yellow}\rule{1.5pt}{2.5ex}}{虚室生白, 吉祥止止}

```

虚室生白, 吉祥止止

\CJKunderanysymbol \CJKunderanysymbol [<选项>] {<深度>} {<符号>} {<文本内容>}

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

例如, 给汉字加三角形, 可以如下设置:

```

1 \CJKunderanysymbol[sep=0.1em]{0.2em}{\tiny$\triangle$}
2 {瞻彼阒者, 虚室生白, \CJKunderline{吉祥止止}}

```


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 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 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 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 LaTeX 2_ε 和 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

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, 用 \makexeCJKinactive 关闭 interchar
\let\origProblemCmd\problemCmd
\renewcommand\problemCmd[1]{\makexeCJKinactive \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 $\text{X}_{\text{Y}}\TeX$ 引擎的支持。

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

应该使用较新版本的 `expl3` 宏包。

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

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

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

判断宏包是否被引入, 可用于文档正文中。

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

下面这些 CJK 系列宏包不应该被使用。

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

将 `CJKfntef` 包替换为 `xeCJKfntef` 包。

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

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

5.2 内部工具

分配临时变量。

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

各种信息函数的缩略形式。

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

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

`\xeCJK_cs_clear:N` 让控制序列的意义为空。

```

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

```

`\xeCJK_swap_cs:NN` 交换 #1 和 #2 的意义。

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

5.3 功能开关

xeCJKActive 事实上, 将开启或关闭 XeTeX 的整个字符类机制。

```

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

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

```

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

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

5.4 字符类别设定

分别用于记录在 xeCJK 中使用的字符类别名称和新建的字符类别的编号。

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

新建一个字符类别。#1 是自定义名称。

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

保存 XeTeX 预定义的字符类别。#1 是自定义名称, #2 是编号。

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

字符类名称对应的控制序列名字。

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

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

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

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

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

Default 369 \xeCJK_save_class:nn { Default } { 0 }
 CJK
 FullLeft XeTeX 0.99994 将字符类总数扩大到 4096¹⁰。
 FullRight 370 \str_const:Nn \c__xeCJK_xetex_version_str
 Boundary 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 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 HangulJamoL、HangulJamoV
 CJStarter 和 HangulJamoT。
 PoZheHao

```

392 \xeCJK_new_class:n { HangulJamo }
393 \xeCJK_new_class:n { HangulJamoL }
394 \xeCJK_new_class:n { HangulJamoV }
395 \xeCJK_new_class:n { HangulJamoT }
396 \xeCJK_new_class:n { CJStarter }
397 \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
398 \clist_const:Nn \c__xeCJK_HalfLeft_chars_clist
399 { "28 , "5B , "60 , "7B , "2329 }
400 \clist_const:Nn \c__xeCJK_HalfRight_chars_clist
401 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , "232A }
402 \clist_const:Nn \c__xeCJK_NormalSpace_chars_clist { "2D , "2F , "5C }
```

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

\c__xeCJK_OP_chars_clist **Open Punctuation (OP)**

U+2018	‘	U+201C	“	U+3008	<	U+300A	《	U+300C	⎵	U+300E	『	U+3010	【
U+3014	〔	U+3016	〔	U+3018	⋈	U+301A	⋈	U+301D	⋈	U+FE17	⎵	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。

```
403 \clist_const:Nn \c__xeCJK_OP_chars_clist
404 {
405   "2018 , "201C ,
406   "3008 , "300A , "300C , "300E , "3010 , "3014 , "3016 , "3018 , "301A , "301D ,
407   "FE17 , "FE35 , "FE37 , "FE39 , "FE3B , "FE3D , "FE3F , "FE41 , "FE43 , "FE47 ,
408   "FE59 , "FE5B , "FE5D , "FF08 , "FF3B , "FF5B , "FF5F , "FF62
409 }
```

\c__xeCJK_PR_chars_clist **Prefix Numeric (PR)**

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

```
410 \clist_const:Nn \c__xeCJK_PR_chars_clist
411 { "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。

```
412 \clist_const:Nn \c__xeCJK_FullLeft_chars_clist
413 {
414   \exp_not:V \c__xeCJK_OP_chars_clist ,
415   \exp_not:V \c__xeCJK_PR_chars_clist
416 }
```

\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。

```
417 \clist_const:Nn \c__xeCJK_CL_chars_clist
418 {
419   "00B7 , "2019 , "201D , "2013 , "2014 , "2025 , "2026 , "2027 , "2E3A ,
420   "3001 , "3002 , "3009 , "300B , "300D , "300F , "3011 , "3015 , "3017 , "3019 ,
421   "301B , "301E , "301F , "FE11 , "FE12 , "FE18 , "FE36 , "FE38 , "FE3A , "FE3C ,
422   "FE3E , "FE40 , "FE42 , "FE44 , "FE48 , "FE50 , "FE52 , "FE5A , "FE5C , "FE5E ,
423   "FF09 , "FF0C , "FF0E , "FF3D , "FF5D , "FF60 , "FF61 , "FF63 , "FF64
424 }
```

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 | ㄣ | U+309D | ㄣ | U+309E | ㄣ | U+30FD | ㄣ | U+30FE | ㄣ |

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。

```

425 \clist_const:Nn \c__xeCJK_hyphens_chars_clist
426 { "301C", "30A0", "FF5E" }
427 \clist_const:Nn \c__xeCJK_iteration_marks_chars_clist
428 { "3005", "303B", "309D", "309E", "30FD", "30FE" }
429 \clist_const:Nn \c__xeCJK_NS_chars_clist
430 { "30FB", "FE54", "FE55", "FF1A", "FF1B", "FF65", "16FE0" }
431 \AtEndOfPackage
432 {
433   \cs_set:Npn \__xeCJK_tmp:w #1
434     { \char_generate:nn {#1} { 12 } }
435   \__xeCJK_add_special_punct:nn { long }
436     { \clist_map_function:NN \c__xeCJK_hyphens_chars_clist \__xeCJK_tmp:w }
437   \cs_undefine:N \__xeCJK_tmp:w
438 }

```

\c__xeCJK_EX_chars_clist Exclamation/Interrogation (EX)

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

```

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

```

\c__xeCJK_IS_chars_clist Infix Numeric Separator (IS)

U+FE10 | ' | U+FE13 | : | U+FE14 | ; |

```

441 \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	ー	U+1B132	☒	U+1B150->"1B152	☒->"1B152	U+1B155	☒	U+1B164->"1B167

```

442 \clist_const:Nn \c__xeCJK_CJ_chars_clist
443 {
444   "3041", "3043", "3045", "3047", "3049", "3063", "3083", "3085", "3087", "308E",
445   "3095", "3096", "30A1", "30A3", "30A5", "30A7", "30A9", "30C3", "30E3", "30E5",
446   "30E7", "30EE", "30F5", "30F6", "30FC", "31F0", "31F1", "31F2", "31F3", "31F4",
447   "31F5", "31F6", "31F7", "31F8", "31F9", "31FA", "31FB", "31FC", "31FD", "31FE",
448   "31FF", "FF67", "FF68", "FF69", "FF6A", "FF6B", "FF6C", "FF6D", "FF6E", "FF6F",
449   "FF70", "1B132", "1B150 -> "1B152", "1B155", "1B164 -> "1B167
450 }

```

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

U+FE6A | % | U+FF05 | % | U+FFE0 | ¢ |

```

451 \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。

```

452 \clist_const:Nc \c__xeCJK_FullRight_chars_clist
453 {
454   \exp_not:V \c__xeCJK_CL_chars_clist ,
455   \exp_not:V \c__xeCJK_NS_chars_clist ,
456   \exp_not:V \c__xeCJK_EX_chars_clist ,
457   \exp_not:V \c__xeCJK_IS_chars_clist ,
458   \exp_not:V \c__xeCJK_PO_chars_clist ,
459   \exp_not:V \c__xeCJK_hyphens_chars_clist
460 }
```

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

```

461 \clist_const:Nc \c__xeCJK_CJK_chars_clist
462 {
```

- 闽南语、客家话阴去和阳去声调标记

```

463   "02EA -> "02EB ,
```

- CJK Radicals Supplement (中日韩部首补充)

```

464   "2E80 -> "2EFF ,
```

- Kangxi Radicals (康熙字典部首)

```

465   "2F00 -> "2FDF ,
```

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

```

466   "2FF0 -> "2FFF ,
```

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

```

467   "3000 -> "303F ,
```

- Hiragana (日文平假名)

```

468   "3040 -> "309F ,
```

- Katakana (日文片假名)

```

469   "30A0 -> "30FF ,
```

- Bopomofo (注音字母)

```

470   "3100 -> "312F ,
```

- Hangul Compatibility Jamo (谚文兼容字母)

```

471   "3130 -> "318F ,
```

- Kanbun (象形字注释标志)

```

472   "3190 -> "319F ,
```

- Bopomofo Extended (注音字母扩展)

```

473   "31A0 -> "31BF ,
```

- CJK Strokes (中日韩笔画)

```

474   "31C0 -> "31EF ,
```

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

```

475   "31F0 -> "31FF ,
```

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

```

476   "3200 -> "32FF ,
```

- CJK Compatibility (中日韩兼容)

```

477   "3300 -> "33FF ,
```

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

478 "3400 -> "4DBF ,

- Yijing Hexagrams Symbols (易经六十四卦符号)

479 "4DC0 -> "4DFF ,

- CJK Unified Ideographs (中日韩统一表意文字)

480 "4E00 -> "9FFF ,

- Yi Syllables (彝文音节)

481 "A000 -> "A48F ,

- Yi Radicals (彝文字根)

482 "A490 -> "A4CF ,

- Hangul Syllables (谚文音节)

483 "AC00 -> "D7AF ,

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

484 "F900 -> "FAFF ,

- Vertical Forms (竖排形式)

485 "FE10 -> "FE1F ,

- CJK Compatibility Forms (中日韩兼容形式)

486 "FE30 -> "FE4F ,

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

487 "FF00 -> "FFEF ,

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

488 "16FE0 -> "16FFF ,

- Tangut (西夏文)

489 "17000 -> "187FF ,

- Tangut Components (西夏文部首)

490 "18800 -> "18AFF ,

- Khitan Small Script (契丹小字)

491 "18B00 -> "18CFF ,

- Tangut Supplement (西夏文增补)

492 "18D00 -> "18D7F ,

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

493 "18D80 -> "18DFF ,

- Kana Extended-B (日文假名扩展 B)

494 "1AFF0 -> "1AFFF ,

- Kana Supplement (日文假名增补)

495 "1B000 -> "1B0FF ,

- Kana Extended-A (日文假名扩展 A)

496 "1B100 -> "1B12F ,

- Small Kana Extension (小型日文假名扩展)

497 "1B130 -> "1B16F ,

- Nushu (女书)

```
498      "1B170 -> "1B2FF ,
```

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

```
499      "1F200 -> "1F2FF ,
```

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

```
500      "20000 -> "2A6DF ,
```

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

```
501      "2A700 -> "2B73F ,
```

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

```
502      "2B740 -> "2B81F ,
```

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

```
503      "2B820 -> "2CEAF ,
```

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

```
504      "2CEB0 -> "2EBEF ,
```

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

```
505      "2EBF0 -> "2EE5F ,
```

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

```
506      "2F800 -> "2FA1F ,
```

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

```
507      "30000 -> "3134F ,
```

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

```
508      "31350 -> "323AF ,
```

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

```
509      "323B0 -> "3347F
```

```
510    }
```

\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

511 \clist_const:Nn \c__xeCJK_CM_chars_clist
512   {
```

- Diacritics (音调符号)

```
513      "302A -> "302F ,
```

- 日文假名浊点

```
514      "3099 -> "309A ,
```

- Variation Selectors (异体字选择符)

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

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

```
516      "E0100 -> "E01EF
517  }
```

\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

```
518 \clist_const:Nn \c_xeCJK_HangulJamoL_chars_clist
519 { "1100 -> "115F , "A960 -> "A97C }
520 \clist_const:Nn \c_xeCJK_HangulJamoV_chars_clist
521 { "1160 -> "11A7 , "D7B0 -> "D7C6 }
522 \clist_const:Nn \c_xeCJK_HangulJamoT_chars_clist
523 { "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。

```
524 \cs_new:Npn \xeCJK_class_num:n #1
525 { \use:c { \_xeCJK_class_csname:n {#1} } }
```

\xeCJKDeclareCharClass 526 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
 527 {
 528 \xeCJK_declare_char_class:nn {#2} {#3}
 529 \IfBooleanT {#1} { \xeCJKResetPunctClass }
 530 }

\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

```
531 \cs_new_protected:Npn \xeCJK_declare_char_class:nn #1#2
532 {
533   \clist_set:Nc \l__xeCJK_tmp_clist {#2}
534   \xeCJK_declare_char_class:nN {#1} \l__xeCJK_tmp_clist
535 }
536 \cs_new_protected:Npn \xeCJK_declare_char_class:nN #1#2
537 {
538   \clist_gconcat:ccN
539   { g__xeCJK_#1_range_clist } { g__xeCJK_#1_range_clist } #2
540   \clist_map_inline:Nn #2
541   {
542     \str_if_eq:nnF {##1} { -> }
543     {
544       \_xeCJK_set_char_class_aux:Nnw \xeCJK_set_char_class:nnn {##1}
545       { \xeCJK_class_num:n {#1} }
546     }
547   }
548   \xeCJK_set_char_class:nnn { "3099 } { "309A } { \xeCJK_class_num:n { CM } }
549 }
550 \NewDocumentCommand \_xeCJK_set_char_class_aux:Nnw
551 { m > { \SplitArgument { 1 } { -> } } m } { #1 #2 }
552 \cs_generate_variant:Nn \clist_gconcat:NNN { cc }
```

_xeCJK_check_num_range:nnNN 553 \cs_new_protected:Npn _xeCJK_check_num_range:nnNN #1#2#3#4
 554 {
 555 \bool_lazy_or:nnTF
 556 { \tl_if_blank_p:n {#1} }
 557 { \tl_if_blank_p:n {#2} }
 558 {
 559 \int_set:Nn #3 { \tl_if_blank:nTF {#1} {#2} {#1} }
 560 \int_set_eq:NN #4 #3
 561 }
 562 {
 563 \int_set:Nn #3 { \int_min:nn {#1} { \tl_if_novalue:nTF {#2} {#1} {#2} } }
 564 \int_set:Nn #4 { \int_max:nn {#1} { \tl_if_novalue:nTF {#2} {#1} {#2} } }
 565 }
 566 }

```

567 \token_if_letter:NF ~~~~ac00
568 {
569   \int_set:Nn \l__xeCJK_begin_int { "AC00 }
570   \int_set:Nn \l__xeCJK_end_int { "D7A3 }
571   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
572   {
573     \char_set_catcode_letter:n { \l__xeCJK_begin_int }
574     \int_incr:N \l__xeCJK_begin_int
575   }
576 }

```

`\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。

```

577 \cs_new_protected:Npn \xeCJK_set_char_class:nnn #1#2#3
578 {
579   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
580   \int_set:Nn \l__xeCJK_tmp_int {#3}
581   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
582   {
583     \tex_XeTeXcharclass:D \l__xeCJK_begin_int = \l__xeCJK_tmp_int
584     \int_incr:N \l__xeCJK_begin_int
585   }
586 }

```

`\__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。

```

587 \cs_new_protected:Npn \__xeCJK_set_char_class_eq:nn #1#2
588 {
589   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#2} }
590   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
591   { \tex_XeTeXcharclass:D ##1 = \l__xeCJK_tmp_int }
592 }

```

`\normalspacedchars` 声明前后不加间距的字符。

```

593 \NewDocumentCommand \normalspacedchars { m }
594 {
595   \tl_map_inline:nn {#1}
596   { \tex_XeTeXcharclass:D `##1 = \xeCJK_class_num:n { NormalSpace } }
597 }

```

`\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。

```

598 \NewDocumentCommand \xeCJKResetPunctClass { }
599 {
600   \clist_gclear:N \g__xeCJK_HalfLeft_range_clist
601   \clist_gclear:N \g__xeCJK_HalfRight_range_clist
602   \clist_gclear:N \g__xeCJK_FullLeft_range_clist
603   \clist_gclear:N \g__xeCJK_FullRight_range_clist
604   \xeCJK_declare_char_class:nN { HalfLeft } \c__xeCJK_HalfLeft_chars_clist
605   \xeCJK_declare_char_class:nN { HalfRight } \c__xeCJK_HalfRight_chars_clist
606   \xeCJK_declare_char_class:nN { FullLeft } \c__xeCJK_FullLeft_chars_clist
607   \xeCJK_declare_char_class:nN { FullRight } \c__xeCJK_FullRight_chars_clist
608   \bool_if:NT \l__xeCJK_pozhehao_ligature_bool
609   { \keys_set:nn { xeCJK / options } { PoZheHaoLigature = true } }
610   \bool_if:NT \l__xeCJK_latin_punct_bool
611   { \keys_set:nn { xeCJK / options } { LatinPunct = true } }
612 }
613 \bool_new:N \l__xeCJK_pozhehao_ligature_bool
614 \bool_new:N \l__xeCJK_latin_punct_bool

```

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

```

615 \NewDocumentCommand \xeCJKResetCharClass { }
616 {

```

```

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

```

设置字符类别。

```
636 \xeCJKResetCharClass
```

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

```

637 \cs_new_protected:Npn \xeCJK_inter_class_toks:nnn #1#2#3
638 {
639   \tex_XeTeXinterchartoks:D \xeCJK_class_num:n {#1} ~
640                               \xeCJK_class_num:n {#2} = {#3}
641 }
642 \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。

```

643 \cs_new:Npn \xeCJK_get_inter_class_toks:nn #1#2
644 {
645   \tex_the:D \tex_XeTeXinterchartoks:D \xeCJK_class_num:n {#1} ~
646                               \xeCJK_class_num:n {#2}
647 }

```

`\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。

```

648 \cs_new_protected:Npn \xeCJK_clear_inter_class_toks:nn #1#2
649 { \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。

```

650 \cs_new_protected:Npn \xeCJK_pre_inter_class_toks:nnn #1#2#3
651 {
652   \xeCJK_inter_class_toks:nne {#1} {#2}
653   { \exp_not:n {#3} \xeCJK_get_inter_class_toks:nn {#1} {#2} }
654 }
655 \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。

```

656 \cs_new_protected:Npn \xeCJK_app_inter_class_toks:nnn #1#2#3
657 {
658   \xeCJK_inter_class_toks:nne {#1} {#2}
659   { \xeCJK_get_inter_class_toks:nn {#1} {#2} \exp_not:n {#3} }
660 }
661 \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。

```

662 \cs_new_protected:Npn \xeCJK_copy_inter_class_toks:nnnn #1#2#3#4
663 {
664   \tl_set:Nc \l__xeCJK_tmp_tl
665     { \xeCJK_get_inter_class_toks:nn {#3} {#4} }
666   \tl_if_empty:NTF \l__xeCJK_tmp_tl
667     {
668       \tl_set:Nc \l__xeCJK_tmp_tl
669         { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
670       \tl_if_empty:NF \l__xeCJK_tmp_tl
671         { \xeCJK_clear_inter_class_toks:nn {#1} {#2} }
672     }
673   { \xeCJK_inter_class_toks:nne {#1} {#2} { \exp_not:o \l__xeCJK_tmp_tl } }
674 }

```

将 #1 和 #2 之间出现的 #3 用 #4 替换。

```

675 \cs_new_protected:Npn \xeCJK_replace_inter_class_toks:nnnn #1#2#3#4
676 {
677   \tl_set:Nc \l__xeCJK_tmp_tl
678     { \xeCJK_get_inter_class_toks:nn {#1} {#2} }
679   \tl_if_empty:NF \l__xeCJK_tmp_tl
680   {
681     \tl_replace_all:Nnn \l__xeCJK_tmp_tl {#3} {#4}
682     \xeCJK_inter_class_toks:nne {#1} {#2}
683     { \exp_not:o \l__xeCJK_tmp_tl }
684   }
685 }

```

清除边界与 CJK 文字、全角左右标点之间的内容。

```

686 \cs_new_protected:Npn \xeCJK_clear_Boundary_and_CJK_toks:
687 { }
688 \cs_new_protected:Npn \__xeCJK_update_clear_toks:n #1
689 {
690   \cs_gset_protected:Npe \xeCJK_clear_Boundary_and_CJK_toks:
691   {
692     \exp_not:o { \xeCJK_clear_Boundary_and_CJK_toks: }
693     \tex_XeTeXinterchartoks:D
694     \xeCJK_class_num:n { Boundary } ~
695     \xeCJK_class_num:n {#1} = { \exp_not:N \prg_do_nothing: }
696   }
697 }

```

保存宏包预先定义的字符类。

```

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

```

```

711 }
712 \clist_map_function:nN
713 {
714   CJK , FullLeft , FullRight , CM , HangulJamo ,
715   HangulJamoL , HangulJamoV , HangulJamoT , CJStarter , PoZheHao
716 }
717 \__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:

```

718 \cs_new_protected:Npn \xeCJK_class_group_begin:
719 {
720   \c_group_begin_token
721   \bool_set_true:N \l__xeCJK_CJK_group_bool
722   \xeCJK_reset_space_factor:
723   \int_zero:N \tex_XeTeXdashbreakstate:D
724 }
725 \bool_new:N \l__xeCJK_CJK_group_bool
726 \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。

```

727 \AtEndOfPackage
728 {
729   \seq_map_inline:Nn \g__xeCJK_class_seq
730   {
731     \str_if_eq:nnTF {#1} { CM }
732     { \xeCJK_copy_inter_class_toks:nnnn { CM } {#1} { CJK } { CJK } }
733     {
734       \xeCJK_copy_inter_class_toks:nnnn { CM } {#1} { CJK } {#1}
735       \str_if_eq:nnF {#1} { CJK }
736       { \xeCJK_copy_inter_class_toks:nnnn {#1} { CM } {#1} { CJK } }
737     }
738   }
739 }

```

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。

```

740 \AtEndOfPackage
741 {
742   \seq_map_inline:Nn \g__xeCJK_class_seq
743   {
744     \str_if_eq:nnF {#1} { HangulJamo }
745     {
746       \xeCJK_copy_inter_class_toks:nnnn { HangulJamo } {#1} { CJK } {#1}
747       \xeCJK_copy_inter_class_toks:nnnn {#1} { HangulJamo } {#1} { CJK }
748     }
749   }
750 }

```

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。

```

751 \AtEndOfPackage
752 {
753   \clist_map_inline:nn { HangulJamoL , HangulJamoV , HangulJamoT }
754   {
755     \seq_map_inline:Nn \g__xeCJK_class_seq
756     {
757       \str_if_eq:nnF {##1} {#1}
758       {
759         \xeCJK_copy_inter_class_toks:nnnn {#1} {##1} { CJK } {##1}
760         \xeCJK_copy_inter_class_toks:nnnn {##1} {#1} {##1} { CJK }
761       }
762     }
763   }
764   \clist_map_inline:nn { HangulJamoL , HangulJamoV , HangulJamoT }
765   {
766     \clist_map_inline:nn { HangulJamoL , HangulJamoV , HangulJamoT }
767     { \xeCJK_copy_inter_class_toks:nnnn {#1} {##1} { CJK } { CJK } }
768   }
769   \xeCJK_clear_inter_class_toks:nn { HangulJamoL } { HangulJamoL }
770   \xeCJK_clear_inter_class_toks:nn { HangulJamoL } { HangulJamoV }
771   \xeCJK_clear_inter_class_toks:nn { HangulJamoV } { HangulJamoV }
772   \xeCJK_clear_inter_class_toks:nn { HangulJamoV } { HangulJamoT }
773   \xeCJK_clear_inter_class_toks:nn { HangulJamoT } { HangulJamoT }
774 }

```

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。

```

775 \AtEndOfPackage
776 {
777   \seq_map_inline:Nn \g__xeCJK_class_seq
778   {
779     \str_if_eq:nnF {#1} { CJStarter }
780     {
781       \xeCJK_copy_inter_class_toks:nnnn { CJStarter } {#1} { CJK } {#1}
782       \xeCJK_copy_inter_class_toks:nnnn {#1} { CJStarter } {#1} { CJK }
783     }
784   }
785   \xeCJK_copy_inter_class_toks:nnnn
786   { CJStarter } { CJStarter } { CJK } { CJK }
787   \xeCJK_pre_inter_class_toks:nnn { CJK } { CJStarter } { \xeCJK_no_break: }
788   \xeCJK_pre_inter_class_toks:nnn
789   { CJStarter } { CJStarter } { \xeCJK_no_break: }
790   \xeCJK_inter_class_toks:nnn { FullRight } { CJStarter }
791   {
792     \xeCJK_FullRight_and_CJStarter:
793     \xeCJK_fallback_symbol:NN
794     \CJKsymbol
795   }
796 }
797 \cs_new_protected:Npn \xeCJK_FullRight_and_CJStarter:
798 {
799   __xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
800   \xeCJK_no_break:
801   __xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
802   __xeCJK_select_font:
803   \CJKglue
804 }

```

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)。

```

805 \AtEndOfPackage

```

```

806 {
807   \seq_map_inline:Nn \g__xeCJK_class_seq
808   {
809     \str_if_eq:nnF {#1} { PoZheHao }
810     {
811       \xeCJK_copy_inter_class_toks:nnnn { PoZheHao } {#1} { FullRight } {#1}
812       \xeCJK_copy_inter_class_toks:nnnn {#1} { PoZheHao } {#1} { FullRight }
813     }
814   }
815 }

816 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
817 {
818   \xeCJK_inter_class_toks:nnn {#1} { CJK }
819   {
820     \__xeCJK_boundary_capture_class:n { CJK }
821     \xeCJK_class_group_begin:
822     \xeCJK_select_font:
823     \xeCJK_clear_inter_class_toks:nn {#1} { CJK }
824     \xeCJK_clear_Boundary_and_CJK_toks:
825     \xeCJK_fallback_symbol:NN
826     \CJKsymbol
827   }
828   \xeCJK_inter_class_toks:nnn { CJK } {#1}
829   {
830     \xeCJK_class_group_end:
831     \__xeCJK_boundary_capture_class:n { default }
832   }
833 }

834 \clist_map_inline:nn { Default , HalfLeft }
835 {
836   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
837   { \xeCJK_Boundary_and_Default: }
838   \xeCJK_app_inter_class_toks:nnn { CJK } {#1}
839   { \CJKecglue }
840 }

```

\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。

```

841 \cs_new_protected:Npn \xeCJK_Boundary_and_Default:
842 {
843   \__xeCJK_boundary_capture_class:n { default }
844   \xeCJK_check_for_ecglue:
845 }
846 \cs_new_protected:Npn \__xeCJK_check_for_xecglue:
847 {
848   \__xeCJK_if_last_glue:TF
849   { \__xeCJK_replace_space: }
850   { \__xeCJK_check_for_ecglue: }
851 }
852 \cs_new_protected:Npn \__xeCJK_check_for_ecglue:
853 {
854   \__xeCJK_if_last_glue:TF
855   { \__xeCJK_recover_ecglue_source_space: }
856   { \__xeCJK_check_for_ecglue_aux: }
857 }

```

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。

```

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

```

```

890     {
891         \dim_compare:nNnTF
892         { \tex_glueshrink:D #1 } > \c_zero_dim
893         { \prg_return_true: }
894         { \prg_return_false: }
895     }
896     { \prg_return_false: }
897 }
898 { \prg_return_false: }
899 }
900 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space:
901 {
902     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
903     {
904         \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
905         \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
906         {
907             \tex_unskip:D
908             \__xeCJK_recover_ecglue_source_space_aux:
909         }
910         { \__xeCJK_recover_ecglue_source_space_fallback: }
911     }
912     { \__xeCJK_recover_ecglue_source_space_fallback: }
913 }
914 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_aux:
915 {
916     \xeCJK_if_last_node:nTF { CJK }
917     { \__xeCJK_recover_ecglue_source_space_success: }
918     {
919         \xeCJK_if_last_node:nTF { CJK-widow }
920         { \__xeCJK_recover_ecglue_source_space_success: }
921         {
922             \xeCJK_if_last_node:nTF { CJK-space }
923             { \__xeCJK_recover_ecglue_source_space_success: }
924             {
925                 \xeCJK_if_last_node:nTF { math }
926                 {
927                     \xeCJK_remove_node:
928                     \__xeCJK_recover_ecglue_source_space_restore:
929                     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
930                 }
931                 {
932                     \__xeCJK_boundary_if_last_math_space:TF
933                     { \__xeCJK_boundary_math_space_consume: }
934                     { \__xeCJK_recover_ecglue_source_space_restore: }
935                 }
936             }
937         }
938     }
939 }
940 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_success:
941 {
942     \xeCJK_remove_node:
943     \__xeCJK_use_ecglue_skip:
944     \bool_if:NT \g__xeCJK_glue_check_pending_bool
945     { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
946 }
947 \cs_new_protected:Npn \__xeCJK_recover_ecglue_source_space_restore:
948 {
949     \skip_horizontal:N \l__xeCJK_last_skip
950 }
951 \cs_new_eq:NN \__xeCJK_recover_ecglue_source_space_fallback: \prg_do_nothing:
952 \cs_new_protected:Npn \__xeCJK_glue_check_expire_stale:
953 {
954     \int_compare:nNnT { \g__xeCJK_boundary_capture_depth_int } = \c_zero_int
955     {
956         \int_compare:nNnT { \tex_lastnodetype:D } < \c_zero_int

```

```

957         { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
958     }
959 }
960 \cs_new_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:

```

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

```

961 \cs_new_protected:Npn \__xeCJK_replace_space:
962 {
963     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
964     \tex_unskip:D
965     \xeCJK_if_last_node:nTF { CJK-space }
966     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
967     {
968         \xeCJK_if_last_node:nTF { CJK }
969         {
970             \skip_if_eq:nnTF
971             { \l__xeCJK_last_skip }
972             { \c_xeCJK_space_skip_tl }
973             { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
974             { \skip_horizontal:N \l__xeCJK_last_skip }
975         }
976         {
977             \__xeCJK_boundary_if_last_math_space:TF
978             {
979                 \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
980                 { \__xeCJK_boundary_math_space_consume: }
981                 { \skip_horizontal:N \l__xeCJK_last_skip }
982             }
983             { \skip_horizontal:N \l__xeCJK_last_skip }
984         }
985     }
986 }
987 \skip_new:N \l__xeCJK_last_skip

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

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

\__xeCJK_Boundary_and_NormalSp: 1008 \cs_new_protected:Npn \xeCJK_Boundary_and_NormalSp:
1009 { \xeCJK_check_for_ecglue_normalsp: }
1010 \cs_new_protected:Npn \__xeCJK_check_for_xecglue_normalsp:
1011 {
1012     \__xeCJK_if_last_glue:TF
1013     { \__xeCJK_replace_space: }
1014     { \__xeCJK_check_for_ecglue_normalsp: }
1015 }
1016 \cs_new_protected:Npn \__xeCJK_check_for_ecglue_normalsp:

```

```

1017 {
1018   \xeCJK_if_last_node:nT { CJK-space }
1019   { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
1020 }
1021 \cs_new_eq:NN \xeCJK_check_for_ecglue_normalsp:
1022   \__xeCJK_check_for_ecglue_normalsp:

1023 \xeCJK_inter_class_toks:nnn { NormalSpace } { Boundary }
1024 {
1025   \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
1026   \peek_meaning_remove:NTF \tex_italiccorrection:D
1027   {
1028     \tex_italiccorrection:D
1029     { \xeCJK_make_node:n { normalspace } }
1030   }
1031   {
1032     \token_if_space:NTF \l_peek_token
1033     { \xeCJK_make_space_node: }
1034     { { \xeCJK_make_node:n { normalspace } } }
1035   }
1036 }

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

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

```

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

```

1064 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip:
1065 {

```

```

1066 \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1067 \__xeCJK_skip_if_interword:NTF \l__xeCJK_last_skip
1068 {
1069   \tex_unskip:D
1070   \__xeCJK_if_last_kern:TF
1071   { \__xeCJK_check_for_glue_skip_kern: }
1072   { \__xeCJK_check_for_glue_skip_nonkern: }
1073 }
1074 { \__xeCJK_check_for_glue_skip_consume_and_fallback: }
1075 }
1076 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_kern:
1077 {
1078   \__xeCJK_boundary_if_last_math_space:TF
1079   {
1080     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1081     { \__xeCJK_boundary_math_space_to_CJK: }
1082     { \__xeCJK_check_for_glue_skip_restore: }
1083   }
1084   { \__xeCJK_check_for_glue_skip_kern_normal: }
1085 }
1086 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_kern_normal:
1087 {
1088   \dim_compare:nNnTF \tex_lastkern:D = { \__xeCJK_node:n { math } }
1089   {
1090     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1091     {
1092       \xeCJK_remove_node:
1093       \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1094       \bool_if:NTF \l__xeCJK_xecglue_bool
1095       { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1096       { \skip_horizontal:N \l__xeCJK_last_skip }
1097     }
1098     { \__xeCJK_check_for_glue_skip_restore: }
1099   }
1100   {
1101     \bool_if:NTF \g__xeCJK_glue_check_pending_bool
1102     {
1103       \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1104       \dim_case:nnF { \tex_lastkern:D }
1105       {
1106         { \__xeCJK_node:n { CJK } }
1107         {
1108           \xeCJK_remove_node:
1109           \bool_if:NTF \l__xeCJK_reserve_space_bool
1110           { \skip_horizontal:N \l__xeCJK_last_skip }
1111           { \CJKglue }
1112         }
1113         { \__xeCJK_node:n { CJK-space } }
1114         { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1115         { \__xeCJK_node:n { CJK-widow } }
1116         {
1117           \xeCJK_remove_node:
1118           \xeCJK_widow_penalty:
1119           \bool_if:NTF \l__xeCJK_reserve_space_bool
1120           { \skip_horizontal:N \l__xeCJK_last_skip }
1121           { \CJKglue }
1122         }
1123       }
1124       { \__xeCJK_check_for_glue_skip_restore: }
1125     }
1126     { \__xeCJK_check_for_glue_skip_restore: }
1127   }
1128 }
1129 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_nonkern:
1130 {
1131   \bool_if:NT \g__xeCJK_glue_check_pending_bool
1132   { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }

```

```

1133   \_xeCJK_if_last_math:TF
1134   {
1135     \bool_if:NTF \l__xeCJK_xecglue_bool
1136     { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1137     { \skip_horizontal:N \l__xeCJK_last_skip }
1138   }
1139   { \_xeCJK_check_for_glue_skip_restore: }
1140 }
1141 \cs_new_protected:Npn \_xeCJK_check_for_glue_skip_restore:
1142 {
1143   \skip_horizontal:N \l__xeCJK_last_skip
1144   \_xeCJK_check_for_glue_auxii:
1145 }
1146 \cs_new_protected:Npn \_xeCJK_check_for_glue_skip_consume_and_fallback:
1147 {
1148   \bool_if:NT \g__xeCJK_glue_check_pending_bool
1149   { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
1150   \_xeCJK_check_for_glue_auxii:
1151 }
1152 \cs_new_protected:Npn \_xeCJK_check_for_glue_auxi:
1153 {
1154   \dim_case:nn { \tex_lastkern:D }
1155   {
1156     { \_xeCJK_node:n { CJK } }
1157     { \xeCJK_remove_node: \CJKglue }
1158     { \_xeCJK_node:n { CJK-space } }
1159     { \xeCJK_remove_node: \_xeCJK_ccglue_or_space: }
1160     { \_xeCJK_node:n { CJK-widow } }
1161     { \xeCJK_remove_node: \xeCJK_widow_penalty: \CJKglue }
1162     { \_xeCJK_node:n { default } }
1163     { \xeCJK_remove_node: \_xeCJK_use_ecglue_skip: }
1164     { \_xeCJK_node:n { math } }
1165     { \xeCJK_remove_node: \_xeCJK_use_ecglue_skip: }
1166     { \_xeCJK_node:n { math-space } }
1167     { \_xeCJK_boundary_math_space_to_CJK: }
1168     { \_xeCJK_node:n { math-space-frozen } }
1169     { \_xeCJK_boundary_math_space_to_CJK: }
1170   }
1171 }
1172 \cs_new_protected:Npn \_xeCJK_check_for_glue_auxii:
1173 {
1174   \xeCJK_if_last_punct:TF
1175   { \_xeCJK_check_for_glue_auxiii: }
1176   { \xeCJK_check_for_xglue: }
1177 }
1178 \cs_new_protected:Npn \_xeCJK_check_for_glue_auxiii:
1179 {
1180   \bool_if:NT \l__xeCJK_last_penalty_bool
1181   { \tex_penalty:D \l__xeCJK_last_penalty_int }
1182   \skip_horizontal:N \l__xeCJK_last_skip
1183   \tl_if_eq:NNF \l__xeCJK_aligni_tl \c__xeCJK_left_tl { \CJKglue }
1184 }
1185 \cs_new_eq:NN \xeCJK_check_for_xglue: \prg_do_nothing:
1186 \cs_new_protected:Npn \_xeCJK_check_for_xglue:
1187 {
1188   \_xeCJK_if_last_glue:T
1189   {
1190     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1191     \tex_unskip:D
1192     \xeCJK_if_last_node:nTF { CJK-space }
1193     { \xeCJK_remove_node: \_xeCJK_ccglue_or_space: }
1194     {
1195       \xeCJK_if_last_node:nTF { default-space }
1196       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1197       { \_xeCJK_check_for_xglue_aux: }
1198     }
1199   }

```

```

1200 }
1201 \cs_new_protected:Npn \__xeCJK_check_for_xglue_aux:
1202 {
1203   \skip_if_eq:nnTF
1204   { \l__xeCJK_last_skip }
1205   { \c_xeCJK_space_skip_tl }
1206   {
1207     \xeCJK_if_last_node:nTF { CJK }
1208     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1209     {
1210       \xeCJK_if_last_node:nTF { default }
1211       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1212       {
1213         \xeCJK_if_last_node:nTF { math }
1214         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1215         {
1216           \__xeCJK_if_last_math:TF
1217           { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1218           { \skip_horizontal:N \l__xeCJK_last_skip }
1219         }
1220       }
1221     }
1222   }
1223   { \skip_horizontal:N \l__xeCJK_last_skip }
1224 }
1225 \cs_new_protected:Npn \__xeCJK_ccglue_or_space:
1226 { \CJKglue }

```

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

__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

```

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

```

```

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

```

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

\xeCJK_make_node:n

```

1259 \cs_new_protected:Npn \xeCJK_declare_node:n #1

```

```

1260 {
1261   \int_gincr:N \g__xeCJK_node_int
1262   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
1263   { \dim_gset:cn } { \dim_const:cn }
1264   { c__xeCJK_#1_node_dim } { \g__xeCJK_node_int sp }
1265 }
1266 \cs_new_protected:Npn \xeCJK_declare_glue_node:n #1
1267 {
1268   \int_gincr:N \g__xeCJK_node_int
1269   \skip_if_exist:cTF { c__xeCJK_#1_node_skip }
1270   { \skip_gset:cn } { \skip_const:cn }
1271   { c__xeCJK_#1_node_skip } { \g__xeCJK_node_int sp }
1272 }
1273 \int_new:N \g__xeCJK_node_int
1274 \int_gset:Nn \g__xeCJK_node_int { 10 }
1275 \tl_new:N \g__xeCJK_last_node_tl
1276 \cs_new_protected:Npn \xeCJK_make_node:n #1
1277 {
1278   \exp_args:Nc \__xeCJK_make_node:N { c__xeCJK_#1_node_dim }
1279   \tl_gset:Nn \g__xeCJK_last_node_tl {#1}
1280 }
1281 \cs_new:Npn \__xeCJK_node:n #1
1282 { \use:c { c__xeCJK_#1_node_dim } }
1283 \cs_new:Npn \__xeCJK_glue_node:n #1
1284 { \use:c { c__xeCJK_#1_node_skip } }
1285 \cs_new_protected:Npn \__xeCJK_make_node:N #1
1286 {
1287   \tex_kern:D - #1
1288   \tex_kern:D #1
1289 }
1290 \cs_new_protected:Npn \xeCJK_remove_node:
1291 {
1292   \__xeCJK_if_last_kern:TF
1293   { \tex_unkern:D \tex_unkern:D }
1294   {
1295     \__xeCJK_if_last_glue:T
1296     { \tex_unskip:D \tex_unskip:D }
1297   }
1298   \tl_gclear:N \g__xeCJK_last_node_tl
1299 }
1300 \xeCJK_declare_node:n { CJK }
1301 \xeCJK_declare_node:n { CJK-space }
1302 \xeCJK_declare_node:n { default }
1303 \xeCJK_declare_node:n { CJK-widow }
1304 \xeCJK_declare_node:n { normalspace }
1305 \xeCJK_declare_glue_node:n { default-space }
1306 \xeCJK_declare_node:n { math }
1307 \xeCJK_declare_node:n { math-space }
1308 \xeCJK_declare_node:n { math-space-frozen }
1309 \bool_new:N \g__xeCJK_glue_check_pending_bool

```

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

__xeCJK_boundary_capture_class:n
 __xeCJK_boundary_if_math_head:n
 __xeCJK_boundary_if_math_tail:n
 __xeCJK_boundary_if_math_tail_space:n
 __xeCJK_boundary_if_math_edge:n
 __xeCJK_boundary_if_last_math_node:n
 __xeCJK_boundary_math_begin:n
 __xeCJK_boundary_math_begin_visible:n
 __xeCJK_boundary_math_end:n
 __xeCJK_boundary_math_end_confirm_node:n
 __xeCJK_boundary_math_end_confirm_space:n
 __xeCJK_boundary_math_end_confirm_space_visible:n
 __xeCJK_boundary_math_end_confirm:n
 __xeCJK_boundary_capture_report_first:n
 __xeCJK_boundary_capture_allocate:n
 __xeCJK_boundary_capture_suspend:n
 __xeCJK_boundary_capture_resume:n
 __xeCJK_boundary_capture_begin:n
 __xeCJK_boundary_inline_box_begin:n

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。

```

1310 \int_new:N \g__xeCJK_boundary_capture_depth_int
1311 \int_new:N \g__xeCJK_boundary_suspend_depth_int
1312 \seq_new:N \g__xeCJK_boundary_active_seq
1313 \prop_new:N \g__xeCJK_boundary_registered_prop
1314 \prop_new:N \g__xeCJK_boundary_reserved_prop
1315 \prop_new:N \g__xeCJK_boundary_user_registered_prop
1316 \seq_new:N \g__xeCJK_boundary_user_applied_seq
1317 \bool_new:N \g__xeCJK_boundary_user_register_closed_bool
1318 \bool_new:N \l__xeCJK_boundary_user_invalid_bool
1319 \bool_new:N \l__xeCJK_boundary_user_mode_given_bool
1320 \tl_new:N \l__xeCJK_boundary_user_name_tl
1321 \tl_new:N \l__xeCJK_boundary_user_strategy_tl
1322 \tl_new:N \l__xeCJK_boundary_user_mode_tl
1323 \tl_new:N \l__xeCJK_boundary_user_value_tl
1324 \seq_new:N \l__xeCJK_boundary_user_value_seq
1325 \box_new:c { g__xeCJK_boundary_capture_1_box }
1326 \box_new:c { g__xeCJK_boundary_capture_2_box }
1327 \box_new:c { g__xeCJK_boundary_capture_3_box }
1328 \box_new:N \l__xeCJK_boundary_post_box
1329 \tl_new:N \l__xeCJK_boundary_node_tl
1330 \tl_new:N \l__xeCJK_boundary_first_tl
1331 \tl_new:N \l__xeCJK_boundary_last_tl
1332 \tl_new:N \l__xeCJK_boundary_active_tl
1333
1334 \cs_new_protected:Npn \__xeCJK_boundary_capture_allocate:n #1
1335 {
1336   \box_if_exist:cF { g__xeCJK_boundary_capture_#1_box }
1337   { \box_new:c { g__xeCJK_boundary_capture_#1_box } }
1338   \tl_if_exist:cF { g__xeCJK_boundary_capture_#1_before_tl }
1339   {
1340     \tl_new:c { g__xeCJK_boundary_capture_#1_before_tl }
1341     \tl_new:c { g__xeCJK_boundary_capture_#1_before_math_space_skip_tl }
1342     \tl_new:c { g__xeCJK_boundary_capture_#1_first_tl }
1343     \tl_new:c { g__xeCJK_boundary_capture_#1_last_tl }
1344     \tl_new:c { g__xeCJK_boundary_capture_#1_kind_tl }
1345     \tl_new:c { g__xeCJK_boundary_capture_#1_space_skip_tl }
1346     \tl_new:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl }
1347     \tl_new:c { g__xeCJK_boundary_capture_#1_ccglue_skip_tl }
1348     \tl_new:c { g__xeCJK_boundary_capture_#1_space_flag_tl }
1349     \tl_new:c { g__xeCJK_boundary_capture_#1_reserve_space_flag_tl }
1350     \tl_new:c { g__xeCJK_boundary_capture_#1_xecglue_flag_tl }
1351     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_flag_tl }
1352     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_space_flag_tl }
1353     \tl_new:c { g__xeCJK_boundary_capture_#1_math_tail_space_skip_tl }
1354   }
1355 }
1356 \__xeCJK_boundary_capture_allocate:n { 1 }
1357 \__xeCJK_boundary_capture_allocate:n { 2 }
1358 \__xeCJK_boundary_capture_allocate:n { 3 }
1359
1360 \cs_new_protected:Npn \__xeCJK_boundary_capture_class:n #1
1361 {
1362   \int_compare:nNnT
1363   { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1364   {
1365     \int_compare:nNnT
1366     { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1367     {
1368       \int_step_inline:nn { \g__xeCJK_boundary_capture_depth_int }

```

```

1369         {
1370             \tl_gset:cn { g__xeCJK_boundary_capture_##1_last_tl } {#1}
1371             \tl_if_empty:cT { g__xeCJK_boundary_capture_##1_first_tl }
1372                 {
1373                     \tl_gset:cn
1374                     { g__xeCJK_boundary_capture_##1_first_tl } {#1}
1375                     \tl_if_eq:cnTF
1376                     { g__xeCJK_boundary_capture_##1_kind_tl } { stream }
1377                     { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1378                     {
1379                         \tl_if_eq:cnT
1380                         { g__xeCJK_boundary_capture_##1_kind_tl } { stream-ulem }
1381                         { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1382                     }
1383                 }
1384             }
1385         }
1386     }
1387 }
1388
1389 \cs_new_protected:Npn \__xeCJK_boundary_math_begin:
1390 {
1391     \mode_if_horizontal:T
1392     {
1393         \scan_stop:
1394         \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1395         \str_case:Vn \l__xeCJK_boundary_node_tl
1396         {
1397             { CJK }
1398             { \CJKecglue }
1399             { CJK-space }
1400             { \xeCJK_space_or_xecglue: }
1401             { CJK-widow }
1402             { \xeCJK_widow_penalty: \CJKecglue }
1403             { default-space }
1404             { \xeCJK_space_glue: }
1405         }
1406     }
1407 }
1408
1409 \cs_new_protected:Npn \__xeCJK_boundary_math_begin_visible:
1410 {
1411     \__xeCJK_boundary_capture_class:n { math }
1412     \__xeCJK_boundary_math_begin:
1413 }
1414
1415 \tl_new:N \l__xeCJK_boundary_math_tl
1416 \tl_new:N \l__xeCJK_boundary_math_type_tl
1417 \skip_new:N \l__xeCJK_boundary_math_skip
1418 \skip_new:N \l__xeCJK_boundary_math_actual_skip
1419 \skip_new:N \l__xeCJK_boundary_math_ecglue_skip
1420 \skip_new:N \l__xeCJK_boundary_math_delta_skip
1421 \dim_new:N \l__xeCJK_boundary_math_space_dim
1422 \dim_new:N \l__xeCJK_boundary_math_ecglue_dim
1423 \dim_new:N \l__xeCJK_boundary_math_stretch_dim
1424 \dim_new:N \l__xeCJK_boundary_math_shrink_dim

```

__xeCJK_boundary_math_set:n 把待判断的参数存入 \l__xeCJK_boundary_math_tl, 并把其中 catcode 4 的 & 换成 \scan_stop:。

触发条件要分清两件事:& 在 L^AT_EX 下默认就是 catcode 4 (latex.ltx), 并非 \halign 把它设成 4; 真正要紧的是**扫描发生的位置**——当 \tl_if_head_eq_meaning:nNTF 等 expl3 条件式在对齐环境内 (eqnarray、align、tabular) 读取参数时, catcode 4 的 & 会终止它正在读的宏参数, 报“Argument of __tl_tl_head:w has an extra }” (#1043)。同一个 token list 在对齐环境之外走同一条件式实测 0 错误, 所以这不是「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参数终止语义。这里只是为语法判断准备副本,实际排版仍使用原始参数,所以替换不影响输出;换成 `\scan_stop:` 而不是直接删除,是为了保住 `&` 的位置,使 `&$x$` 这类参数仍被正确判定为「首项不是公式」。

两条范围说明,避免后人误以为漏改:

- `\tl_replace_all:Nvn` 不进入嵌套花括号组, `{ a { x & y } b }` 只得到 `a{x&y}b`。这是够用的:只有顶层对齐符会破坏 `expl3` 条件式的括号组扫描(实测组内含 `catcode-4 &` 时,裸 `\tl_if_head_eq_meaning:nNTF`、`\tl_reverse:N`、`\tl_head:n`、`\tl_tail:n` 均不报错),而 `\__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,下一层的 `&` 到那时才需要消解。
- `\__xeCJK_boundary_if_math_tail_space:n` 不走本函数,因为它只做 `\tl_trim_right_spaces:n` 与 `\tl_if_eq:nNTF`,不做 `head/tail` 记号扫描,参数含 `catcode-4 &` 时实测 0 错误。需要消解的只有做扫描的 `\__xeCJK_boundary_if_math_head:n` 与 `\__xeCJK_boundary_if_math_tail:n` 两个入口。

`\tl_replace_all:Nnn` 及其变体(这里用的是 `\tl_replace_all:Nvn`)按 `catcode` 与 `charcode` 双重相等匹配,所以模式里的 `&` 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 **LaTeX 环境默认把 `&` 设为 4**(`latex.ltx: \catcode\&=4`),而 `\ExplSyntaxOn` 不改 38(实测它把 9、32 设为 9,58、95 设为 11,126 设为 10,而 38 前后都是 4)。注意 `\c_code_cctab` 之类的 `catcode` 表常量与此无关:它只在被 `\cctab_select:N` 选中时才生效,本包从不选它。

关键在于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 `\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。真正的风险是加载期的 régime:若 `\usepackage` 之前外层做过 `\catcode\&=12` (或在某个非默认 `catcode` 表下 `\input` 本包),字面 `{ & }` 读进来就不是对齐符,此后一律静默失配。`\c__xeCJK_alignment_tl` 由 `\char_set_catcode_alignment:N` 在局部组里现场生成,与读取时的 régime 无关,防住的正是这一种情形。

```

1425 \group_begin:
1426   \char_set_catcode_alignment:N \&
1427   \tl_const:Nn \c__xeCJK_alignment_tl { & }
1428 \group_end:
1429
1430 \cs_new_protected:Npn \__xeCJK_boundary_math_set:n #1
1431 {
1432   \tl_set:Nc \l__xeCJK_boundary_math_tl
1433     { \tl_trim_spaces:n {#1} }
1434   \tl_replace_all:Nvn \l__xeCJK_boundary_math_tl
1435     \c__xeCJK_alignment_tl { \scan_stop: }
1436 }
1437
1438 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_head:n #1
1439 { T , F , TF }
1440 {
1441   \__xeCJK_boundary_math_set:n {#1}
1442   \tl_if_empty:NTF \l__xeCJK_boundary_math_tl
1443     { \prg_return_false: }
1444     {
1445       \exp_args:NV \tl_if_head_is_group:nTF
1446         \l__xeCJK_boundary_math_tl
1447         {
1448           \exp_args:Nc \__xeCJK_boundary_if_math_head:nTF
1449             { \tl_head:V \l__xeCJK_boundary_math_tl }
1450             { \prg_return_true: }
1451             { \prg_return_false: }
1452         }
1453       {
1454         \exp_args:NV \tl_if_head_eq_meaning:nNTF
1455           \l__xeCJK_boundary_math_tl $

```

```

1456         { \prg_return_true: }
1457     {
1458         \exp_args:NV \tl_if_head_eq_meaning:nNTF
1459         \l__xeCJK_boundary_math_tl \{
1460         { \prg_return_true: }
1461         {
1462             \exp_args:NV \tl_if_head_eq_meaning:nNTF
1463             \l__xeCJK_boundary_math_tl \ensuremath
1464             { \prg_return_true: }
1465             { \prg_return_false: }
1466         }
1467     }
1468 }
1469 }
1470 }
1471
1472 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_tail:n #1
1473 { T , F , TF }
1474 {
1475     \__xeCJK_boundary_math_set:n {#1}
1476     \tl_reverse:N \l__xeCJK_boundary_math_tl
1477     \exp_args:NV \__xeCJK_boundary_if_math_tail_reversed:nTF
1478     \l__xeCJK_boundary_math_tl
1479     { \prg_return_true: }
1480     { \prg_return_false: }
1481 }

```

`\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,先识别 `\ensuremath` 的参数,否则递归检查组内末尾。所有语法判断都只产生候选:未知宏也可能把末尾的分组、\$ 或 \) 当作参数消费, `\__xeCJK_boundary_math_end:n` 还要用当前列表的实际 `math` 节点或 `marker` 确认,才会发布末类别。

```

1482 \prg_new_protected_conditional:Npnn
1483 \__xeCJK_boundary_if_math_tail_reversed:n #1 { T , F , TF }
1484 {
1485     \tl_if_empty:nTF {#1}
1486     { \prg_return_false: }
1487     {
1488         \tl_if_head_eq_meaning:nNTF {#1} $
1489         { \prg_return_true: }
1490         {
1491             \tl_if_head_eq_meaning:nNTF {#1} \)
1492             { \prg_return_true: }
1493             {
1494                 \tl_if_head_is_group:nTF {#1}
1495                 {
1496                     \tl_set:Ne \l__xeCJK_boundary_math_tl
1497                     { \tl_trim_left_spaces:e { \tl_tail:n {#1} } }
1498                     \exp_args:NV \tl_if_head_eq_meaning:nNTF
1499                     \l__xeCJK_boundary_math_tl \ensuremath
1500                     { \prg_return_true: }
1501                     {
1502                         \exp_args:Ne \__xeCJK_boundary_if_math_tail:nTF
1503                         { \tl_head:n {#1} }
1504                         { \prg_return_true: }
1505                         { \prg_return_false: }
1506                     }
1507                 }
1508                 { \prg_return_false: }
1509             }
1510         }
1511     }
1512 }
1513
1514 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_edge:n #1
1515 { T , F , TF }
1516 {

```

```

1517 \__xeCJK_boundary_if_math_head:nTF {#1}
1518 { \prg_return_true: }
1519 {
1520 \__xeCJK_boundary_if_math_tail:nTF {#1}
1521 { \prg_return_true: }
1522 { \prg_return_false: }
1523 }
1524 }
1525
1526 \cs_new_protected:Npn \__xeCJK_boundary_math_begin:n #1
1527 {
1528 \__xeCJK_boundary_if_math_head:nT {#1}
1529 { \__xeCJK_boundary_math_begin_visible: }
1530 }
1531
1532 \cs_new_protected:Npn \__xeCJK_boundary_math_end:n #1
1533 {
1534 \__xeCJK_boundary_if_math_tail:nT {#1}
1535 {
1536 \__xeCJK_boundary_if_math_tail_space:nTF {#1}
1537 { \__xeCJK_boundary_math_end_confirm_space: }
1538 { \__xeCJK_boundary_math_end_confirm_node: }
1539 }
1540 }
1541
1542 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_math_tail_space:n #1
1543 { T , F , TF }
1544 {
1545 \tl_set:Nn \l__xeCJK_boundary_math_tl {#1}
1546 \tl_set:Nn \l__xeCJK_boundary_node_tl
1547 { \tl_trim_right_spaces:n {#1} }
1548 \tl_if_eq:NNTF \l__xeCJK_boundary_math_tl \l__xeCJK_boundary_node_tl
1549 { \prg_return_false: }
1550 { \prg_return_true: }
1551 }
1552
1553 \prg_new_conditional:Npnn \__xeCJK_boundary_if_last_math_node: { T , F , TF }
1554 {
1555 \__xeCJK_if_last_math:TF
1556 { \prg_return_true: }
1557 {
1558 \xeCJK_if_last_node:nTF { math }
1559 { \prg_return_true: }
1560 { \prg_return_false: }
1561 }
1562 }
1563
1564 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_node:
1565 {
1566 \__xeCJK_boundary_if_last_math_node:T
1567 { \__xeCJK_boundary_math_end_confirm: }
1568 }

```

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。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 结束时按普通 stream 或冻结空格分别处理。

```

1569 \cs_new_protected:Npn \__xeCJK_boundary_math_end_confirm_space:
1570 {
1571 \__xeCJK_if_last_glue:T
1572 {
1573 \skip_set_eq:NN \l__xeCJK_boundary_math_skip \tex_lastskip:D
1574 \__xeCJK_skip_if_interword:NT \l__xeCJK_boundary_math_skip
1575 {
1576 \tex_unskip:D
1577 \__xeCJK_boundary_if_last_math_node:T

```

```

1578         {
1579             \_xeCJK_boundary_math_end_confirm_space_visible:
1580         }
1581         \skip_horizontal:N \l__xeCJK_boundary_math_skip
1582     }
1583 }
1584 }
1585
1586 \cs_new_protected:Npn \_xeCJK_boundary_math_end_confirm_space_visible:
1587 {
1588     \_xeCJK_boundary_math_end_confirm:
1589     \tl_gset:cn
1590     {
1591         g__xeCJK_boundary_capture_
1592         \int_use:N \g__xeCJK_boundary_capture_depth_int
1593         _math_tail_space_flag_tl
1594     }
1595     { true }
1596     \tl_if_eq:cnT
1597     {
1598         g__xeCJK_boundary_capture_
1599         \int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl
1600     }
1601     { stream }
1602     {
1603         \tl_gset:ce
1604         {
1605             g__xeCJK_boundary_capture_
1606             \int_use:N \g__xeCJK_boundary_capture_depth_int
1607             _math_tail_space_skip_tl
1608         }
1609         { \skip_use:N \l__xeCJK_boundary_math_skip }
1610     }
1611 }
1612
1613 \cs_new_protected:Npn \_xeCJK_boundary_math_end_confirm:
1614 {
1615     \int_compare:nNtT
1616     { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1617     {
1618         \int_compare:nNtT
1619         { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1620         {
1621             \tl_gset:cn
1622             {
1623                 g__xeCJK_boundary_capture_
1624                 \int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl
1625             }
1626             { math }
1627             \tl_gset:cn
1628             {
1629                 g__xeCJK_boundary_capture_
1630                 \int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
1631             }
1632             { true }
1633             \tl_gset:cn
1634             {
1635                 g__xeCJK_boundary_capture_
1636                 \int_use:N \g__xeCJK_boundary_capture_depth_int
1637                 _math_tail_space_flag_tl
1638             }
1639             { false }
1640         }
1641     }
1642 }
1643
1644 \cs_new_protected:Npn \_xeCJK_boundary_capture_report_first:n #1

```

```

1645 {
1646   \int_compare:nNtT
1647     { \g__xeCJK_boundary_capture_depth_int } > \c_zero_int
1648     {
1649       \int_compare:nNtT
1650         { \g__xeCJK_boundary_suspend_depth_int } = \c_zero_int
1651         {
1652           \int_step_inline:nn { \g__xeCJK_boundary_capture_depth_int }
1653           {
1654             \tl_if_empty:cT { g__xeCJK_boundary_capture_##1_first_tl }
1655             {
1656               \tl_gset:cn
1657                 { g__xeCJK_boundary_capture_##1_first_tl } {#1}
1658               \tl_if_eq:cnTF
1659                 { g__xeCJK_boundary_capture_##1_kind_tl } { stream }
1660                 { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1661                 {
1662                   \tl_if_eq:cnT
1663                     { g__xeCJK_boundary_capture_##1_kind_tl } { stream-ulem }
1664                     { \__xeCJK_boundary_capture_emit_left:nn {##1} {#1} }
1665                 }
1666             }
1667           }
1668         }
1669       }
1670     }
1671
1672 \cs_new_protected:Npn \__xeCJK_boundary_capture_emit_left:nn #1#2
1673 {
1674   \tl_set:Nc \l__xeCJK_boundary_node_tl
1675     { \tl_use:c { g__xeCJK_boundary_capture_#1_before_tl } }
1676   \__xeCJK_boundary_emit_left:nVn
1677     {#1} \l__xeCJK_boundary_node_tl {#2}
1678   \tl_gset:cn
1679     { g__xeCJK_boundary_capture_#1_space_flag_tl } { false }
1680 }
1681
1682 \cs_new_protected:Npn \__xeCJK_boundary_capture_suspend:
1683 {
1684   \int_gincr:N \g__xeCJK_boundary_suspend_depth_int
1685   \tl_if_exist:cF
1686     {
1687       g__xeCJK_boundary_suspend_\int_use:N
1688       \g__xeCJK_boundary_suspend_depth_int _node_tl
1689     }
1690     {
1691       \tl_new:c
1692         {
1693           g__xeCJK_boundary_suspend_\int_use:N
1694           \g__xeCJK_boundary_suspend_depth_int _node_tl
1695         }
1696       \tl_new:c
1697         {
1698           g__xeCJK_boundary_suspend_\int_use:N
1699           \g__xeCJK_boundary_suspend_depth_int _pending_tl
1700         }
1701     }
1702   \exp_args:Nc \tl_gset_eq:NN
1703     {
1704       g__xeCJK_boundary_suspend_\int_use:N
1705       \g__xeCJK_boundary_suspend_depth_int _node_tl
1706     }
1707   \g__xeCJK_last_node_tl
1708   \tl_gset:ce
1709     {
1710       g__xeCJK_boundary_suspend_\int_use:N
1711       \g__xeCJK_boundary_suspend_depth_int _pending_tl

```

```

1712     }
1713     { \bool_if:NTF \g__xeCJK_glue_check_pending_bool { true } { false } }
1714 }
1715
1716 \cs_new_protected:Npn \__xeCJK_boundary_capture_resume:
1717 {
1718     \tl_gset:Nx \g__xeCJK_last_node_tl
1719     {
1720         \tl_use:c
1721         {
1722             g__xeCJK_boundary_suspend_\int_use:N
1723             \g__xeCJK_boundary_suspend_depth_int _node_tl
1724         }
1725     }
1726     \tl_if_eq:cnTF
1727     {
1728         g__xeCJK_boundary_suspend_\int_use:N
1729         \g__xeCJK_boundary_suspend_depth_int _pending_tl
1730     }
1731     { true }
1732     { \bool_gset_true:N \g__xeCJK_glue_check_pending_bool }
1733     { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
1734     \int_gdecr:N \g__xeCJK_boundary_suspend_depth_int
1735 }
1736
1737 \cs_new_protected:Npn \__xeCJK_boundary_inline_box_begin:
1738 {
1739     \int_compare:nNnTF
1740     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1741     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1742     {
1743         \mode_if_math:TF
1744         { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1745         {
1746             \mode_leave_vertical:
1747             \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1748             \__xeCJK_boundary_box_begin:
1749         }
1750     }
1751 }
1752
1753 \cs_new_protected:Npn \__xeCJK_boundary_inline_box_end:n #1
1754 {
1755     \seq_gpop:NN
1756     \g__xeCJK_boundary_active_seq
1757     \l__xeCJK_boundary_active_tl
1758     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1759     { \__xeCJK_boundary_box_end:n {#1} }
1760 }
1761
1762 \cs_new_protected:Npn \__xeCJK_boundary_inline_last_box_begin:
1763 {
1764     \int_compare:nNnTF
1765     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1766     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1767     {
1768         \mode_if_math:TF
1769         { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1770         {
1771             \mode_leave_vertical:
1772             \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1773             \__xeCJK_boundary_capture_begin:
1774             \tl_gset:cn
1775             {
1776                 g__xeCJK_boundary_capture_\int_use:N
1777                 \g__xeCJK_boundary_capture_depth_int _kind_tl
1778             }

```

```

1779         { box }
1780     }
1781 }
1782 }
1783
1784 \cs_new_protected:Npn \__xeCJK_boundary_inline_last_box_end:n #1
1785 {
1786     \seq_gpop:NN
1787     \g__xeCJK_boundary_active_seq
1788     \l__xeCJK_boundary_active_tl
1789     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1790     { \__xeCJK_boundary_last_box_end:n {#1} }
1791 }
1792
1793 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_begin:n #1
1794 { \__xeCJK_boundary_inline_stream_begin:nn {#1} { stream } }
1795
1796 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_begin:nn #1#2
1797 {
1798     \int_compare:nNnTF
1799     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1800     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1801     {
1802         \mode_if_math:TF
1803         { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1804         {
1805             \mode_leave_vertical:
1806             \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1807             \__xeCJK_boundary_capture_begin:
1808             \tl_gset:cn
1809             {
1810                 g__xeCJK_boundary_capture\_int\_use:N
1811                 \g__xeCJK_boundary_capture_depth_int\_kind\_tl
1812             }
1813             {#2}
1814             \str_case:nn {#1}
1815             {
1816                 { default } { \__xeCJK_boundary_capture_class:n { default } }
1817                 { first-default } { \__xeCJK_boundary_capture_class:n { default } }
1818             }
1819         }
1820     }
1821 }
1822
1823 \cs_new_protected:Npn \__xeCJK_boundary_inline_stream_end:n #1
1824 {
1825     \seq_gpop:NN
1826     \g__xeCJK_boundary_active_seq
1827     \l__xeCJK_boundary_active_tl
1828     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1829     {
1830         \scan_stop:
1831         \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1832         \tl_set:Ne \l__xeCJK_boundary_first_tl
1833         {
1834             \tl_use:c
1835             {
1836                 g__xeCJK_boundary_capture\_int\_use:N
1837                 \g__xeCJK_boundary_capture_depth_int\_first\_tl
1838             }
1839         }
1840         \tl_set:Ne \l__xeCJK_boundary_last_tl
1841         {
1842             \tl_use:c
1843             {
1844                 g__xeCJK_boundary_capture\_int\_use:N
1845                 \g__xeCJK_boundary_capture_depth_int\_last\_tl

```

```

1846         }
1847     }

```

流式命令若在最外层留下 xeCJK marker, 就以该 marker 记录的类别更新末类别。这适用于内部使用不可见定界字符、最后又把正文末尾 marker 移到外层列表的命令; 没有末尾 marker 的 annotation/math 仍使用 capture 观察值。

```

1848         \exp_args:NV
1849         \__xeCJK_boundary_set_last_from_node:n \l__xeCJK_boundary_node_tl
1850         \__xeCJK_boundary_set_math_tail_from_capture:

```

default 的开始 hook 已经固定首类别; 这里仅覆写末类别即可让 stream 两端都为 Default, 同时不丢弃其他模式实际观察到的首类别。

```

1851         \str_if_eq:nnT {#1} { default }
1852         { \tl_set:Nn \l__xeCJK_boundary_last_tl {#1} }
1853         \tl_gclear:N \g__xeCJK_last_node_tl
1854         \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
1855         \tl_if_empty:NTF \l__xeCJK_boundary_first_tl
1856         {
1857             \__xeCJK_boundary_replay_before:
1858         }
1859         {
1860             \tl_if_empty:NT \l__xeCJK_boundary_last_tl
1861             { \tl_set_eq:NN \l__xeCJK_boundary_last_tl \l__xeCJK_boundary_first_tl }
1862             \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_last_tl
1863         }
1864         \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
1865     }
1866 }
1867
1868 \cs_new_protected:Npn \__xeCJK_boundary_hmode_transparent_begin:
1869 {
1870     \int_compare:nNnTF
1871     { \g__xeCJK_boundary_suspend_depth_int } > \c_zero_int
1872     { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1873     {
1874         \mode_if_horizontal:TF
1875         {
1876             \seq_gpush:Nn \g__xeCJK_boundary_active_seq { true }
1877             \__xeCJK_boundary_capture_begin:
1878             \tl_gset:cn
1879             {
1880                 g__xeCJK_boundary_capture_\int_use:N
1881                 \g__xeCJK_boundary_capture_depth_int _kind_tl
1882             }
1883             { transparent }
1884         }
1885         { \seq_gpush:Nn \g__xeCJK_boundary_active_seq { false } }
1886     }
1887 }
1888
1889 \cs_new_protected:Npn \__xeCJK_boundary_hmode_transparent_end:
1890 {
1891     \seq_gpop:NN
1892     \g__xeCJK_boundary_active_seq
1893     \l__xeCJK_boundary_active_tl
1894     \tl_if_eq:NnT \l__xeCJK_boundary_active_tl { true }
1895     {
1896         \__xeCJK_boundary_replay_before:
1897         \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
1898     }
1899 }
1900
1901 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent:
1902 {
1903     \mode_if_horizontal:T

```

```

1904 {
1905   \tl_if_empty:NF \g__xeCJK_last_node_tl
1906   {
1907     \__xeCJK_if_last_hlist:TF
1908     {
1909       \box_set_to_last:N \l__xeCJK_boundary_post_box
1910       \dim_compare:nNnTF
1911       {
1912         \box_wd:N \l__xeCJK_boundary_post_box
1913         + \box_ht:N \l__xeCJK_boundary_post_box
1914         + \box_dp:N \l__xeCJK_boundary_post_box
1915       }
1916       = { \c_zero_dim }
1917       { \__xeCJK_boundary_post_transparent_relocate: }
1918       { \box_use_drop:N \l__xeCJK_boundary_post_box }
1919     }
1920     {
1921       \xeCJK_if_last_node:TF
1922       {
1923         \exp_args:NV
1924         \__xeCJK_boundary_replay_node:n \g__xeCJK_last_node_tl
1925       }
1926       { }
1927     }
1928   }
1929 }
1930 }

```

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 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末尾 glue, 再按节点证据检查其下方的 marker。命中后把“marker + glue”整体移到盒子后面, 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 时按原顺序还原, 不能把任意 glue 搬过 hbox。普通 stream 的 math-space 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。只有有限阶的候选 glue 才可能是被盒子遮住的源码空格或参数内空格; \hfill 等无限阶 (fil/fill) 填充 glue 不是候选, 保持 marker、glue、盒子的原有次序 (#1085, 见 __xeCJK_boundary_post_transparent_relocate_glue:)。

```

1931 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent_relocate:
1932 {
1933   \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1934   \tl_if_empty:NTF \l__xeCJK_boundary_node_tl
1935   {
1936     \__xeCJK_if_last_glue:TF
1937     { \__xeCJK_boundary_post_transparent_relocate_glue: }
1938     { \box_use_drop:N \l__xeCJK_boundary_post_box }
1939   }
1940   {

```

盒子直接盖住 marker, 中间没有 glue。

```

1941     \box_use_drop:N \l__xeCJK_boundary_post_box
1942     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
1943     { \__xeCJK_boundary_math_space_expire: }
1944     {
1945       \exp_args:NV
1946       \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
1947     }
1948   }
1949 }
1950
1951 \cs_new_protected:Npn \__xeCJK_boundary_post_transparent_relocate_glue:
1952 {
1953   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1954   \skip_if_finite:nTF \l__xeCJK_last_skip

```

```

1955 {
1956   \tex_unskip:D
1957   \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
1958   \tl_if_empty:NTF \l__xeCJK_boundary_node_tl
1959   {
1960     \skip_horizontal:N \l__xeCJK_last_skip
1961     \box_use_drop:N \l__xeCJK_boundary_post_box
1962   }
1963   {
1964     \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
1965     {

```

`math-space` 过期时不再移动后缀;先恢复候选 `glue`,再放回零尺寸盒子,保持与含同一盒子的直接公式相同的节点顺序。

```

1966       \skip_horizontal:N \l__xeCJK_last_skip
1967       \box_use_drop:N \l__xeCJK_boundary_post_box
1968       \__xeCJK_boundary_math_space_expire:
1969     }
1970     {
1971       \box_use_drop:N \l__xeCJK_boundary_post_box
1972       \exp_args:NV
1973       \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
1974       \skip_horizontal:N \l__xeCJK_last_skip
1975     }
1976   }
1977 }

```

只有有限阶(finite)的候选 `glue` 才可能是被零尺寸盒子遮住的源码空格或参数内空格(后者宽度不一定等于词间空格、也不一定带 `shrink`, 例如 `math-space` 场景里的显式 `\hskip`), 因此这里只排除 `\hfill/\hfil` 一类无限阶(fil/fill)填充 `glue`, 不能收紧成 `\__xeCJK_skip_if_interword:N`(它要求带 `shrink` 且宽度等于词间空格, 会误伤 `math-space` 的显式 `glue`)。无限阶 `glue` 不是任何源码空格候选:不移动它,直接把零尺寸盒子放回末尾,保持 `marker`、`glue`、盒子的原有次序。否则会把 `\null` 前的填充 `glue` 误搬到盒子之后,令 `\null` 不在列表末尾,破坏两侧 `\hfill` 的对称(#1085)。

```

1978   { \box_use_drop:N \l__xeCJK_boundary_post_box }
1979 }
1980
1981 \cs_new_protected:Npn \__xeCJK_boundary_capture_begin:
1982 {
1983   \int_gincr:N \g__xeCJK_boundary_capture_depth_int
1984   \int_compare:nNnT
1985     { \g__xeCJK_boundary_capture_depth_int } > { 2 }
1986     {
1987       \exp_args:Ne \__xeCJK_boundary_capture_allocate:n
1988         { \int_use:N \g__xeCJK_boundary_capture_depth_int }
1989     }
1990   \tl_gclear:c
1991   {
1992     g__xeCJK_boundary_capture_\int_use:N
1993     \g__xeCJK_boundary_capture_depth_int _before_tl
1994   }
1995   \tl_gclear:c
1996   {
1997     g__xeCJK_boundary_capture_\int_use:N
1998     \g__xeCJK_boundary_capture_depth_int _before_math_space_skip_tl
1999   }
2000   \tl_gclear:c
2001   {
2002     g__xeCJK_boundary_capture_\int_use:N
2003     \g__xeCJK_boundary_capture_depth_int _first_tl
2004   }
2005   \tl_gclear:c
2006   {
2007     g__xeCJK_boundary_capture_\int_use:N

```

```

2008     \g__xeCJK_boundary_capture_depth_int _last_tl
2009   }
2010   \tl_gclear:c
2011   {
2012     g__xeCJK_boundary_capture_\int_use:N
2013     \g__xeCJK_boundary_capture_depth_int _kind_tl
2014   }

```

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 只把已求值的 skip 数值序列化到本层 tl, 不展开原始/ 定义。

```

2015   \xeCJK_glue_to_skip:nN
2016   { \tex_kern:D \c_zero_dim \xeCJK_space_glue: }
2017   \l__xeCJK_tmp_skip
2018   \tl_gset:ce
2019   {
2020     g__xeCJK_boundary_capture_\int_use:N
2021     \g__xeCJK_boundary_capture_depth_int _space_skip_tl
2022   }
2023   { \skip_use:N \l__xeCJK_tmp_skip }
2024   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_tmp_skip
2025   \tl_gset:ce
2026   {
2027     g__xeCJK_boundary_capture_\int_use:N
2028     \g__xeCJK_boundary_capture_depth_int _ecglue_skip_tl
2029   }
2030   { \skip_use:N \l__xeCJK_tmp_skip }
2031   \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_tmp_skip
2032   \tl_gset:ce
2033   {
2034     g__xeCJK_boundary_capture_\int_use:N
2035     \g__xeCJK_boundary_capture_depth_int _ccglue_skip_tl
2036   }
2037   { \skip_use:N \l__xeCJK_tmp_skip }
2038   \tl_gset:cn
2039   {
2040     g__xeCJK_boundary_capture_\int_use:N
2041     \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2042   }
2043   { false }
2044   \tl_gset:ce
2045   {
2046     g__xeCJK_boundary_capture_\int_use:N
2047     \g__xeCJK_boundary_capture_depth_int _reserve_space_flag_tl
2048   }
2049   { \bool_if:NTF \l__xeCJK_reserve_space_bool { true } { false } }
2050   \tl_gset:ce
2051   {
2052     g__xeCJK_boundary_capture_\int_use:N
2053     \g__xeCJK_boundary_capture_depth_int _xecglue_flag_tl
2054   }
2055   { \bool_if:NTF \l__xeCJK_xecglue_bool { true } { false } }
2056   \tl_gset:cn
2057   {
2058     g__xeCJK_boundary_capture_\int_use:N
2059     \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
2060   }
2061   { false }
2062   \tl_gset:cn
2063   {
2064     g__xeCJK_boundary_capture_\int_use:N
2065     \g__xeCJK_boundary_capture_depth_int _math_tail_space_flag_tl
2066   }
2067   { false }
2068   \tl_gclear:c
2069   {
2070     g__xeCJK_boundary_capture_\int_use:N

```

```

2071     \g__xeCJK_boundary_capture_depth_int _math_tail_space_skip_tl
2072   }
2073   \__xeCJK_boundary_capture_space:
2074   \scan_stop:
2075   \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2076   \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { math-space }
2077   {
2078     \__xeCJK_if_last_glue:TF
2079     {
2080       \tl_gset:ce
2081       {
2082         g__xeCJK_boundary_capture_
2083         \int_use:N \g__xeCJK_boundary_capture_depth_int
2084         _before_math_space_skip_tl
2085       }
2086       { \skip_use:N \tex_lastskip:D }
2087     }
2088     {
2089       \tl_gset:ce
2090       {
2091         g__xeCJK_boundary_capture_
2092         \int_use:N \g__xeCJK_boundary_capture_depth_int
2093         _before_math_space_skip_tl
2094       }
2095       { \skip_use:N \l__xeCJK_boundary_math_actual_skip }
2096     }
2097   }
2098   \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { CJK-space }
2099   {
2100     \tl_gset:cn
2101     {
2102       g__xeCJK_boundary_capture_\int_use:N
2103       \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2104     }
2105     { true }
2106   }
2107   \tl_if_empty:NT \l__xeCJK_boundary_node_tl
2108   {
2109     \tl_if_eq:cnT
2110     {
2111       g__xeCJK_boundary_capture_\int_use:N
2112       \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2113     }
2114     { true }
2115     {
2116       \__xeCJK_boundary_restore_space:
2117       \tl_gset:cn
2118       {
2119         g__xeCJK_boundary_capture_\int_use:N
2120         \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2121       }
2122       { false }
2123     }
2124   }
2125   \exp_args:Nc \tl_gset_eq:NN
2126   {
2127     g__xeCJK_boundary_capture_\int_use:N
2128     \g__xeCJK_boundary_capture_depth_int _before_tl
2129   }
2130   \l__xeCJK_boundary_node_tl
2131   \tl_gclear:N \g__xeCJK_last_node_tl
2132   \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2133 }
2134
2135 \cs_new_protected:Npn \__xeCJK_boundary_box_begin:
2136 {
2137   \__xeCJK_boundary_capture_begin:

```

```

2138 \tl_gset:cn
2139 {
2140   g__xeCJK_boundary_capture_\int_use:N
2141   \g__xeCJK_boundary_capture_depth_int _kind_tl
2142 }
2143 { box }
2144 \exp_args:Nc \__xeCJK_boundary_box_open:N
2145 {
2146   g__xeCJK_boundary_capture_\int_use:N
2147   \g__xeCJK_boundary_capture_depth_int _box
2148 }
2149 }
2150
2151 \cs_new_protected:Npn \__xeCJK_boundary_box_open:N #1
2152 {
2153   \tex_global:D \tex_setbox:D #1
2154   \tex_hbox:D \c_group_begin_token
2155 }
2156
2157 \cs_new_protected:Npn \__xeCJK_boundary_capture_space:
2158 {
2159   \__xeCJK_if_last_glue:T
2160   {
2161     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
2162     \__xeCJK_skip_if_interword:NT \l__xeCJK_last_skip
2163     {
2164       \tl_gset:ce
2165       {
2166         g__xeCJK_boundary_capture_\int_use:N
2167         \g__xeCJK_boundary_capture_depth_int _space_skip_tl
2168       }
2169       { \skip_use:N \l__xeCJK_last_skip }
2170       \tl_gset:cn
2171       {
2172         g__xeCJK_boundary_capture_\int_use:N
2173         \g__xeCJK_boundary_capture_depth_int _space_flag_tl
2174       }
2175       { true }
2176       \tex_unskip:D
2177     }
2178   }
2179 }
2180
2181 \cs_new_protected:Npn \__xeCJK_boundary_pop_node:N #1
2182 {
2183   \tl_clear:N #1
2184   \xeCJK_if_last_node:TF
2185   {
2186     \dim_case:nnF { \l__xeCJK_last_kern_dim }
2187     {
2188       { \__xeCJK_node:n { CJK } } { \tl_set:Nn #1 { CJK } }
2189       { \__xeCJK_node:n { CJK-space } } { \tl_set:Nn #1 { CJK-space } }
2190       { \__xeCJK_node:n { CJK-widow } } { \tl_set:Nn #1 { CJK-widow } }
2191       { \__xeCJK_node:n { default } } { \tl_set:Nn #1 { default } }
2192       { \__xeCJK_node:n { math } } { \tl_set:Nn #1 { math } }
2193       { \__xeCJK_node:n { math-space } }
2194       {
2195         \tl_set:Nn #1 { math-space }
2196         \__xeCJK_boundary_math_space_decode:
2197       }
2198       { \__xeCJK_node:n { math-space-frozen } }
2199       { \tl_set:Nn #1 { math-space-frozen } }
2200       { \__xeCJK_node:n { normalspace } } { \tl_set:Nn #1 { normalspace } }
2201     }
2202     { \__xeCJK_make_node:N \l__xeCJK_last_kern_dim }
2203   }
2204   {

```

```

2205     \xeCJK_if_last_node:nT { default-space }
2206     {
2207         \xeCJK_remove_node:
2208         \tl_set:Nn #1 { default-space }
2209     }
2210 }
2211 }
2212
2213 \prg_new_conditional:Npnn \__xeCJK_boundary_if_last_math_space: { T , F , TF }
2214 {
2215     \xeCJK_if_last_node:nTF { math-space }
2216     { \prg_return_true: }
2217     {
2218         \xeCJK_if_last_node:nTF { math-space-frozen }
2219         { \prg_return_true: }
2220         { \prg_return_false: }
2221     }
2222 }
2223
2224 \cs_new_protected:Npn \__xeCJK_boundary_math_space_consume:
2225 {
2226     \xeCJK_if_last_node:nTF { math-space }
2227     {
2228         \xeCJK_remove_node:
2229         \__xeCJK_boundary_math_space_decode:
2230     }
2231     { \xeCJK_remove_node: }
2232     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2233 }

```

math-space 后面的两对零净宽 kern 依次保存普通 stream 空格的伸长量和收缩量。marker 位于最外侧, 移除 marker 后按相反次序取回两对 kern。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。

```

2234 \cs_new_protected:Npn \__xeCJK_boundary_math_space_set_actual:n #1
2235 {
2236     \skip_set:Nn \l__xeCJK_boundary_math_actual_skip {#1}
2237     \dim_set:Nn \l__xeCJK_boundary_math_stretch_dim
2238     { \tex_gluestretch:D \l__xeCJK_boundary_math_actual_skip }
2239     \dim_set:Nn \l__xeCJK_boundary_math_shrink_dim
2240     { \tex_glueshrink:D \l__xeCJK_boundary_math_actual_skip }
2241     \skip_set:Nn \l__xeCJK_boundary_math_actual_skip
2242     {
2243         \c_zero_dim
2244         plus \l__xeCJK_boundary_math_stretch_dim
2245         minus \l__xeCJK_boundary_math_shrink_dim
2246     }
2247 }
2248
2249 \cs_new_protected:Npn \__xeCJK_boundary_math_space_encode:
2250 {
2251     \dim_set:Nn \l__xeCJK_boundary_math_stretch_dim
2252     { \tex_gluestretch:D \l__xeCJK_boundary_math_actual_skip }
2253     \dim_set:Nn \l__xeCJK_boundary_math_shrink_dim
2254     { \tex_glueshrink:D \l__xeCJK_boundary_math_actual_skip }
2255     \__xeCJK_make_node:N \l__xeCJK_boundary_math_stretch_dim
2256     \__xeCJK_make_node:N \l__xeCJK_boundary_math_shrink_dim
2257 }
2258
2259 \cs_new_protected:Npn \__xeCJK_boundary_math_space_decode:
2260 {
2261     \dim_zero:N \l__xeCJK_boundary_math_stretch_dim
2262     \dim_zero:N \l__xeCJK_boundary_math_shrink_dim
2263     \xeCJK_if_last_node:TF
2264     {
2265         \dim_set_eq:NN
2266         \l__xeCJK_boundary_math_shrink_dim \l__xeCJK_last_kern_dim

```

```

2267     \xeCJK_if_last_node:TF
2268     {
2269         \dim_set_eq:NN
2270         \l__xeCJK_boundary_math_stretch_dim \l__xeCJK_last_kern_dim
2271     }
2272     { }
2273 }
2274 { }
2275 \skip_set:Nn \l__xeCJK_boundary_math_actual_skip
2276 {
2277     \c_zero_dim
2278     plus \l__xeCJK_boundary_math_stretch_dim
2279     minus \l__xeCJK_boundary_math_shrink_dim
2280 }
2281 }

```

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计算,伸缩差额则扣除参数内实际空格的伸缩量。`box`、`wrapped-box` 和 `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,外层只能扣除词间空格的自然宽度,并完整保留 `CJKecglue` 自己的伸缩分量。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自身的固有宽度。

```

2282 \cs_new_protected:Npn \__xeCJK_boundary_math_space_set_delta:n #1
2283 {
2284     \str_if_eq:nnTF {#1} { math-space-frozen }
2285     {
2286         \dim_set:Nn \l__xeCJK_boundary_math_space_dim
2287         { \l__xeCJK_boundary_math_skip }
2288         \dim_set:Nn \l__xeCJK_boundary_math_ecglue_dim
2289         { \l__xeCJK_boundary_math_ecglue_skip }
2290         \dim_set:Nn \l__xeCJK_tmp_dim
2291         {
2292             \dim_max:nn
2293             {
2294                 \l__xeCJK_boundary_math_ecglue_dim
2295                 - \l__xeCJK_boundary_math_space_dim
2296             }
2297             { \c_zero_dim }
2298         }
2299         \skip_set:Nn \l__xeCJK_boundary_math_delta_skip
2300         {
2301             \l__xeCJK_boundary_math_ecglue_skip
2302             - \l__xeCJK_boundary_math_ecglue_dim
2303             + \l__xeCJK_tmp_dim
2304         }
2305     }
2306     {
2307         \dim_set:Nn \l__xeCJK_boundary_math_space_dim
2308         { \l__xeCJK_boundary_math_skip }
2309         \dim_set:Nn \l__xeCJK_boundary_math_ecglue_dim
2310         { \l__xeCJK_boundary_math_ecglue_skip }
2311         \dim_set:Nn \l__xeCJK_tmp_dim
2312         {
2313             \l__xeCJK_boundary_math_ecglue_dim
2314             - \l__xeCJK_boundary_math_space_dim
2315         }
2316         \skip_set:Nn \l__xeCJK_boundary_math_delta_skip
2317         {
2318             \l__xeCJK_boundary_math_ecglue_skip
2319             - \l__xeCJK_boundary_math_ecglue_dim
2320             + \l__xeCJK_tmp_dim
2321             - \l__xeCJK_boundary_math_actual_skip
2322         }
2323     }
2324 }
2325

```

```

2326 \cs_new_protected:Npn \__xeCJK_boundary_math_space_to_CJK:
2327 {
2328   \xeCJK_if_last_node:nTF { math-space-frozen }
2329   { \tl_set:Nn \l__xeCJK_boundary_math_type_tl { math-space-frozen } }
2330   { \tl_set:Nn \l__xeCJK_boundary_math_type_tl { math-space } }
2331   \__xeCJK_boundary_math_space_consume:
2332   \bool_if:NT \l__xeCJK_xecglue_bool
2333   {
2334     \skip_set:Nn \l__xeCJK_boundary_math_skip
2335     { \c_xeCJK_space_skip_tl }
2336     \skip_set_eq:NN
2337     \l__xeCJK_boundary_math_ecglue_skip \l__xeCJK_ecglue_skip
2338     \exp_args:NV \__xeCJK_boundary_math_space_set_delta:n
2339     \l__xeCJK_boundary_math_type_tl
2340     \skip_if_eq:nnF
2341     { \l__xeCJK_boundary_math_delta_skip } { \c_zero_skip }
2342     { \skip_horizontal:N \l__xeCJK_boundary_math_delta_skip }
2343   }
2344 }
2345
2346 \cs_new_protected:Npn \__xeCJK_boundary_box_end:n #1
2347 {
2348   \c_group_end_token
2349   \__xeCJK_boundary_box_finish:nn {#1} { wrapped }
2350 }
2351
2352 \cs_new_protected:Npn \__xeCJK_boundary_last_box_end:n #1
2353 {
2354   \__xeCJK_if_last_hlist:TF
2355   {
2356     \exp_args:Nc \box_set_to_last:N
2357     {
2358       g__xeCJK_boundary_capture\int_use:N
2359       \g__xeCJK_boundary_capture_depth_int _box
2360     }
2361     \__xeCJK_boundary_box_finish:nn {#1} { last }
2362   }
2363   {
2364     \tl_gclear:N \g__xeCJK_last_node_tl
2365     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2366     \__xeCJK_boundary_replay_before:
2367     \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
2368   }
2369 }
2370
2371 \int_new:N \g__xeCJK_boundary_probe_type_int
2372 \box_new:N \l__xeCJK_boundary_probe_box
2373 \tl_new:N \l__xeCJK_boundary_probe_name_tl
2374 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_capture_box_sized:
2375 { T , F , TF }
2376 {
2377   \tl_set:Nc \l__xeCJK_boundary_probe_name_tl
2378   {
2379     g__xeCJK_boundary_capture_
2380     \int_use:N \g__xeCJK_boundary_capture_depth_int _box
2381   }
2382   \bool_lazy_and:nnTF
2383   {
2384     ! \dim_compare_p:nNn
2385     { \box_wd:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2386   }
2387   {
2388     \bool_lazy_or_p:nn
2389     {
2390       ! \dim_compare_p:nNn
2391       { \box_ht:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2392     }

```

```

2393     {
2394         ! \dim_compare_p:nNn
2395         { \box_dp:c { \l__xeCJK_boundary_probe_name_tl } } = \c_zero_dim
2396     }
2397 }
2398 { \prg_return_true: }
2399 { \prg_return_false: }
2400 }
2401 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_capture_box_visible:
2402 { T , F , TF }
2403 {
2404     \__xeCJK_boundary_if_capture_box_sized:TF
2405     {

```

只复制节点, 不重扫 token, 无排版副作用。

```

2406     \hbox_set:Nn \l__xeCJK_boundary_probe_box
2407     {
2408         \hbox_unpack:c { \l__xeCJK_boundary_probe_name_tl }
2409         \int_gset:Nn \g__xeCJK_boundary_probe_type_int
2410         { \tex_lastnodetype:D }
2411     }
2412     \box_clear:N \l__xeCJK_boundary_probe_box

```

\lastnodetype 编码沿用 eTeX 手册的节点类型表 (0 传统 char、1 hlist、3 rule、9 whatsit、10 math、11 glue、12 kern); XeTeX 把 native 文本实现为 word 节点、同样报 9, 但它必然已被 interchar 观察, 不会进入这里。这里把 char、rule、math, 以及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。

```

2413     \int_case:nnTF { \g__xeCJK_boundary_probe_type_int }
2414     {
2415         { 0 } { }
2416         { 3 } { }
2417         { 10 } { }
2418         { 12 } { }
2419     }
2420     { \prg_return_true: }
2421     { \prg_return_false: }
2422 }
2423 { \prg_return_false: }
2424 }
2425
2426 \tl_new:N \g__xeCJK_boundary_probe_last_tl
2427 \cs_new_protected:Npn \__xeCJK_boundary_box_set_last_from_node:
2428 {
2429     \tl_set:Nc \l__xeCJK_boundary_probe_name_tl
2430     {
2431         \g__xeCJK_boundary_capture_
2432         \int_use:N \g__xeCJK_boundary_capture_depth_int _box
2433     }
2434     \tl_gclear:N \g__xeCJK_boundary_probe_last_tl
2435     \hbox_set:Nn \l__xeCJK_boundary_probe_box
2436     {
2437         \hbox_unpack:c { \l__xeCJK_boundary_probe_name_tl }
2438         \__xeCJK_boundary_pop_node:N \l__xeCJK_boundary_node_tl
2439         \tl_if_eq:NnT \l__xeCJK_boundary_node_tl { math-space }
2440         { \tl_set:Nn \l__xeCJK_boundary_node_tl { math-space-frozen } }
2441         \tl_gset_eq:NN
2442         \g__xeCJK_boundary_probe_last_tl \l__xeCJK_boundary_node_tl
2443     }
2444     \box_clear:N \l__xeCJK_boundary_probe_box
2445     \exp_args:NV
2446     \__xeCJK_boundary_set_last_from_node:n \g__xeCJK_boundary_probe_last_tl
2447 }
2448
2449 \cs_new_protected:Npn \__xeCJK_boundary_box_finish:nn #1#2
2450 {

```

```

2451 \tl_set:Nc \l__xeCJK_boundary_first_tl
2452 {
2453   \tl_use:c
2454   {
2455     g__xeCJK_boundary_capture_\int_use:N
2456     \g__xeCJK_boundary_capture_depth_int _first_tl
2457   }
2458 }
2459 \tl_set:Nc \l__xeCJK_boundary_last_tl
2460 {
2461   \tl_use:c
2462   {
2463     g__xeCJK_boundary_capture_\int_use:N
2464     \g__xeCJK_boundary_capture_depth_int _last_tl
2465   }
2466 }
2467 \tl_gclear:N \g__xeCJK_last_node_tl
2468 \bool_gset_false:N \g__xeCJK_glue_check_pending_bool

```

嵌套 box 结束时写入的 marker 会留在当前盒子的节点列表末尾。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推断直接写进所有外层 capture。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使观察值不准,此时应以节点列表末尾的 marker 为准。

```

2469 \__xeCJK_boundary_box_set_last_from_node:
2470 \__xeCJK_boundary_set_math_tail_from_capture:

```

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的首尾类别;stream 路径则由开始 hook 固定首类别、结束 hook 固定末类别,两条路径的公开语义相同。

```

2471 \str_if_eq:nnT {#1} { default }
2472 {
2473   \tl_set:Nn \l__xeCJK_boundary_first_tl { default }
2474   \tl_set:Nn \l__xeCJK_boundary_last_tl { default }
2475 }

```

rule 和尚未由公式观察器记录的 math 不触发 interchar 转换。若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),却没有观察到任何类别,就按 Default 首尾重建边界。空 makebox、strut 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,仍按无可见输出处理(issue 998)。只把 default 报告给尚未取得首类别的外层 capture,使 \nfss@text 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节点列表末尾由本层写入的 marker,并据此更新末类别。

```

2476 \tl_if_empty:NT \l__xeCJK_boundary_first_tl
2477 {
2478   \__xeCJK_boundary_if_capture_box_visible:T
2479   {
2480     \tl_set:Nn \l__xeCJK_boundary_first_tl { default }
2481     \tl_set:Nn \l__xeCJK_boundary_last_tl { default }
2482     \__xeCJK_boundary_capture_report_first:n { default }
2483   }
2484 }
2485 \tl_if_empty:NTF \l__xeCJK_boundary_first_tl
2486 { \__xeCJK_boundary_box_end_transparent:n {#2} }
2487 {
2488   \tl_set:Nc \l__xeCJK_boundary_node_tl
2489   {
2490     \tl_use:c
2491     {
2492       g__xeCJK_boundary_capture_\int_use:N
2493       \g__xeCJK_boundary_capture_depth_int _before_tl
2494     }
2495   }
2496   \exp_args:NVV \__xeCJK_boundary_emit_left:nn

```

```

2497         \l__xeCJK_boundary_node_tl
2498         \l__xeCJK_boundary_first_tl
2499     \box_use_drop:c
2500     {
2501         g__xeCJK_boundary_capture_\int_use:N
2502         \g__xeCJK_boundary_capture_depth_int _box
2503     }
2504     \tl_if_empty:NT \l__xeCJK_boundary_last_tl
2505     { \tl_set_eq:NN \l__xeCJK_boundary_last_tl \l__xeCJK_boundary_first_tl }
2506     \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_last_tl
2507 }
2508 \int_gdecr:N \g__xeCJK_boundary_capture_depth_int
2509 }
2510
2511 \cs_new_protected:Npn \__xeCJK_boundary_box_end_transparent:n #1
2512 {
2513     \str_if_eq:nnTF {#1} { wrapped }
2514     {
2515         \hbox_unpack_drop:c
2516         {
2517             g__xeCJK_boundary_capture_\int_use:N
2518             \g__xeCJK_boundary_capture_depth_int _box
2519         }
2520     }
2521     {
2522         \box_use_drop:c
2523         {
2524             g__xeCJK_boundary_capture_\int_use:N
2525             \g__xeCJK_boundary_capture_depth_int _box
2526         }
2527     }
2528     \__xeCJK_boundary_replay_before:
2529 }
2530
2531 \cs_new_protected:Npn \__xeCJK_boundary_replay_before:
2532 {
2533     \tl_set:Nc \l__xeCJK_boundary_node_tl
2534     {
2535         \tl_use:c
2536         {
2537             g__xeCJK_boundary_capture_
2538             \int_use:N \g__xeCJK_boundary_capture_depth_int _before_tl
2539         }
2540     }
2541     \clist_if_in:nVTF
2542     { math-space , math-space-frozen } \l__xeCJK_boundary_node_tl
2543     {
2544         \tl_if_eq:NnTF \l__xeCJK_boundary_node_tl { math-space }
2545         { \__xeCJK_boundary_replay_before_math_space: }
2546         { \__xeCJK_boundary_replay_node:n { math-space-frozen } }
2547     }
2548     {
2549         \exp_args:NV \__xeCJK_boundary_replay_node:n \l__xeCJK_boundary_node_tl
2550         \__xeCJK_boundary_restore_space:
2551     }
2552 }

```

普通 stream 的 `math-space` 只在真实参数空格仍紧邻列表末尾时重放。`transparent` 命令若已经在二者之间写入 `special`、零尺寸盒子等节点, 直接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, 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。

```

2553 \cs_new_protected:Npn \__xeCJK_boundary_replay_before_math_space:
2554 {
2555     \__xeCJK_if_last_glue:T
2556     {
2557         \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D

```

```

2558     \skip_if_eq:nnT
2559     { \l__xeCJK_last_skip }
2560     {
2561         \tl_use:c
2562         {
2563             g__xeCJK_boundary_capture_
2564             \int_use:N \g__xeCJK_boundary_capture_depth_int
2565             _before_math_space_skip_tl
2566         }
2567     }
2568     {
2569         \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2570         {
2571             \tl_use:c
2572             {
2573                 g__xeCJK_boundary_capture_
2574                 \int_use:N \g__xeCJK_boundary_capture_depth_int
2575                 _before_math_space_skip_tl
2576             }
2577         }
2578         \__xeCJK_boundary_replay_node:n { math-space }
2579     }
2580 }
2581 }
2582
2583 \cs_new_protected:Npn \__xeCJK_boundary_math_space_expire:
2584 {
2585     \tl_gclear:N \g__xeCJK_last_node_tl
2586     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
2587 }
2588
2589 \cs_new_protected:Npn \__xeCJK_boundary_emit_left:nn #1#2
2590 {
2591     \__xeCJK_boundary_emit_left:nnn
2592     { \int_use:N \g__xeCJK_boundary_capture_depth_int } {#1} {#2}
2593 }
2594
2595 \cs_new_protected:Npn \__xeCJK_boundary_emit_left:nnn #1#2#3
2596 {
2597     \clist_if_in:nnTF { math-space , math-space-frozen } {#2}
2598     { \__xeCJK_boundary_emit_left_math_space:nnn {#1} {#2} {#3} }
2599     { \__xeCJK_boundary_emit_left_normal:nnn {#1} {#2} {#3} }
2600 }
2601 \cs_new_protected:Npn \__xeCJK_boundary_emit_left_math_space:nnn #1#2#3
2602 {
2603     \str_if_eq:nnT {#3} { CJK }
2604     {
2605         \tl_if_eq:cnT
2606         { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2607         {
2608             \str_if_eq:nnT {#2} { math-space }
2609             {
2610                 \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2611                 {
2612                     \tl_use:c
2613                     { g__xeCJK_boundary_capture_#1_before_math_space_skip_tl }
2614                 }
2615             }
2616             \skip_set:Nn \l__xeCJK_boundary_math_skip
2617             { \tl_use:c { g__xeCJK_boundary_capture_#1_space_skip_tl } }
2618             \skip_set:Nn \l__xeCJK_boundary_math_ecglue_skip
2619             { \tl_use:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl } }
2620             \__xeCJK_boundary_math_space_set_delta:n {#2}
2621             \skip_if_eq:nnF
2622             { \l__xeCJK_boundary_math_delta_skip } { \c_zero_skip }
2623             {
2624                 \__xeCJK_boundary_use_glue:nn {#1}

```

```

2625         { \l__xeCJK_boundary_math_delta_skip }
2626     }
2627 }
2628 }
2629 }
2630 \cs_new_protected:Npn \__xeCJK_boundary_emit_left_normal:nnn #1#2#3
2631 {
2632     \clist_if_in:nnTF { CJK , CJK-space , CJK-widow } {#2}
2633     {
2634         \str_if_eq:nnTF {#3} { math }
2635         {
2636             \tl_if_eq:cnTF
2637             { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2638             {
2639                 \tl_if_eq:cnTF
2640                 { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2641                 { \__xeCJK_boundary_use_ecglue:n {#1} }
2642                 { \__xeCJK_boundary_restore_space:n {#1} }
2643             }
2644             { \__xeCJK_boundary_use_ecglue:n {#1} }
2645         }
2646     {
2647         \str_if_eq:nnTF {#3} { CJK }
2648         {
2649             \str_if_eq:nnT {#2} { CJK-widow }
2650             { \xeCJK_widow_penalty: }
2651             \tl_if_eq:cnTF
2652             { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2653             {
2654                 \tl_if_eq:cnTF
2655                 { g__xeCJK_boundary_capture_#1_reserve_space_flag_tl }
2656                 { true }
2657                 { \__xeCJK_boundary_restore_space:n {#1} }
2658                 { \__xeCJK_boundary_use_ccglue:n {#1} }
2659             }
2660             { \__xeCJK_boundary_use_ccglue:n {#1} }
2661         }
2662         { \__xeCJK_boundary_use_ecglue:n {#1} }
2663     }
2664 }
2665 {
2666     \clist_if_in:nnTF { default , default-space , math } {#2}
2667     {
2668         \str_if_eq:nnTF {#3} { CJK }
2669         {
2670             \tl_if_eq:cnTF
2671             { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2672             {
2673                 \tl_if_eq:cnTF
2674                 { g__xeCJK_boundary_capture_#1_xecglue_flag_tl } { true }
2675                 { \__xeCJK_boundary_use_ecglue:n {#1} }
2676                 { \__xeCJK_boundary_restore_space:n {#1} }
2677             }
2678             { \__xeCJK_boundary_use_ecglue:n {#1} }
2679         }
2680         { \__xeCJK_boundary_restore_space:n {#1} }
2681     }
2682     { \__xeCJK_boundary_restore_space:n {#1} }
2683 }
2684 }
2685 \cs_generate_variant:Nn \__xeCJK_boundary_emit_left:nnn { nVn }
2686
2687 \cs_new_protected:Npn \__xeCJK_boundary_use_ecglue:n #1
2688 {
2689     \__xeCJK_boundary_use_glue:nn {#1}
2690     { \tl_use:c { g__xeCJK_boundary_capture_#1_ecglue_skip_tl } }
2691 }

```

```

2692
2693 \cs_new_protected:Npn \__xeCJK_boundary_use_ccglue:n #1
2694 {
2695   \__xeCJK_boundary_use_glue:nn {#1}
2696   { \tl_use:c { g__xeCJK_boundary_capture_#1_ccglue_skip_tl } }
2697 }
2698
2699 \cs_new_protected:Npn \__xeCJK_boundary_use_glue:nn #1#2
2700 {
2701   \tl_if_eq:cnTF
2702   { g__xeCJK_boundary_capture_#1_kind_tl } { stream-ulem }
2703   { \use:c { __xeCJK_boundary_use_ulem_glue:n } {#2} }
2704   { \skip_horizontal:n {#2} }
2705 }
2706
2707 \cs_new_protected:Npn \__xeCJK_boundary_restore_space:
2708 {
2709   \__xeCJK_boundary_restore_space:n
2710   { \int_use:N \g__xeCJK_boundary_capture_depth_int }
2711 }
2712
2713 \cs_new_protected:Npn \__xeCJK_boundary_restore_space:n #1
2714 {
2715   \tl_if_eq:cnT
2716   { g__xeCJK_boundary_capture_#1_space_flag_tl } { true }
2717   {
2718     \__xeCJK_boundary_use_glue:nn {#1}
2719     { \tl_use:c { g__xeCJK_boundary_capture_#1_space_skip_tl } }
2720   }
2721 }
2722
2723 \cs_new_protected:Npn \__xeCJK_boundary_set_last_from_node:n #1
2724 {
2725   \clist_if_in:nnTF { CJK , CJK-space , CJK-widow } {#1}
2726   { \tl_set:Nn \l__xeCJK_boundary_last_tl { CJK } }
2727   {
2728     \clist_if_in:nnT
2729     { default , default-space } {#1}
2730     { \tl_set:Nn \l__xeCJK_boundary_last_tl { default } }
2731     \str_if_eq:nnT {#1} { math }
2732     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math } }
2733     \str_if_eq:nnT {#1} { math-space }
2734     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space } }
2735     \str_if_eq:nnT {#1} { math-space-frozen }
2736     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space-frozen } }
2737   }
2738 }
2739
2740 \cs_new_protected:Npn \__xeCJK_boundary_set_math_tail_from_capture:
2741 {
2742   \tl_if_eq:cnT
2743   {
2744     g__xeCJK_boundary_capture_
2745     \int_use:N \g__xeCJK_boundary_capture_depth_int _math_tail_flag_tl
2746   }
2747   { true }
2748   {
2749     \tl_if_eq:cnTF
2750     {
2751       g__xeCJK_boundary_capture_
2752       \int_use:N \g__xeCJK_boundary_capture_depth_int
2753       _math_tail_space_flag_tl
2754     }
2755     { true }
2756     {
2757       \tl_if_eq:cnTF
2758       {

```

```

2759         g__xeCJK_boundary_capture_
2760         \int_use:N \g__xeCJK_boundary_capture_depth_int _kind_tl
2761     }
2762     { stream }
2763     {
2764         \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space }
2765         \exp_args:Ne \__xeCJK_boundary_math_space_set_actual:n
2766         {
2767             \tl_use:c
2768             {
2769                 g__xeCJK_boundary_capture_
2770                 \int_use:N \g__xeCJK_boundary_capture_depth_int
2771                 _math_tail_space_skip_tl
2772             }
2773         }
2774         { \tl_set:Nn \l__xeCJK_boundary_last_tl { math-space-frozen } }
2775     }
2776     { \tl_set:Nn \l__xeCJK_boundary_last_tl { math } }
2777 }
2778 }
2779 }

```

Default 字符在盒子内部更新的 spacefactor 不会传播到外层列表; stream 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 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 错用于盒外以 1000 生成的源码空格。

```

2780 \cs_new_protected:Npn \__xeCJK_boundary_replay_node:n #1
2781 {
2782     \clist_if_in:nnT
2783     {
2784         default , default-space , math , math-space , math-space-frozen ,
2785         normalspace
2786     }
2787     {#1}
2788     { \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D }
2789     \str_if_eq:nnTF {#1} { math-space }
2790     {
2791         \__xeCJK_boundary_math_space_encode:
2792         \xeCJK_make_node:n {#1}
2793         \__xeCJK_boundary_enable_source_space:n {#1}
2794     }
2795     {
2796         \str_if_eq:nnTF {#1} { math-space-frozen }
2797         {
2798             \xeCJK_make_node:n {#1}
2799             \__xeCJK_boundary_enable_source_space:n {#1}
2800         }
2801         {
2802             \str_if_eq:nnTF {#1} { default-space }
2803             {
2804                 \__xeCJK_make_space_node:
2805                 \tl_gset:Nn \g__xeCJK_last_node_tl { default }
2806             }
2807             {
2808                 \clist_if_in:nnT
2809                 { CJK , CJK-space , CJK-widow , default , math , normalspace }
2810                 {#1}
2811                 {
2812                     \xeCJK_make_node:n {#1}
2813                     \__xeCJK_boundary_enable_source_space:n {#1}
2814                 }
2815             }
2816         }
2817     }
2818 }
2819 \cs_new_protected:Npn \__xeCJK_boundary_enable_source_space:n #1
2820 {

```

```

2821 \clist_if_in:nnT
2822 {
2823   CJK , CJK-space , CJK-widow , math , math-space ,
2824   math-space-frozen
2825 }
2826 {#1}
2827 { \bool_gset_true:N \g__xeCJK_glue_check_pending_bool }
2828 }

2829 \cs_new_protected:Npn \__xeCJK_boundary_register_box:nn #1#2
2830 {
2831   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2832   {
2833     \prop_gput:Nnn \g__xeCJK_boundary_registered_prop {#1} {#2}
2834     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2835     { \__xeCJK_boundary_inline_last_box_begin: }
2836     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2837     { \__xeCJK_boundary_inline_last_box_end:n {#2} }
2838   }
2839 }
2840
2841 \cs_new_protected:Npn \__xeCJK_boundary_register_wrapped_box:nn #1#2
2842 {
2843   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2844   {
2845     \prop_gput:Nnn
2846     \g__xeCJK_boundary_registered_prop {#1} { wrapped-box/#2 }
2847     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2848     { \__xeCJK_boundary_inline_box_begin: }
2849     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2850     { \__xeCJK_boundary_inline_box_end:n {#2} }
2851   }
2852 }
2853
2854 \cs_new_protected:Npn \__xeCJK_boundary_register_transparent:n #1
2855 {
2856   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2857   {
2858     \prop_gput:Nnn
2859     \g__xeCJK_boundary_registered_prop {#1} { transparent }
2860     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2861     { \__xeCJK_boundary_hmode_transparent_begin: }
2862     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2863     { \__xeCJK_boundary_hmode_transparent_end: }
2864   }
2865 }
2866
2867 \cs_new_protected:Npn \__xeCJK_boundary_register_post_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} { post-transparent }
2873     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2874     { \__xeCJK_boundary_post_transparent: }
2875   }
2876 }
2877
2878 \cs_new_protected:Npn \__xeCJK_boundary_register_stream:nn #1#2
2879 {
2880   \prop_if_in:NnF \g__xeCJK_boundary_registered_prop {#1}
2881   {
2882     \prop_gput:Nnn \g__xeCJK_boundary_registered_prop {#1} { stream/#2 }
2883     \AddToHook { cmd / #1 / before } [ xeCJK / boundary ]
2884     { \__xeCJK_boundary_inline_stream_begin:n {#2} }
2885     \AddToHook { cmd / #1 / after } [ xeCJK / boundary ]
2886     { \__xeCJK_boundary_inline_stream_end:n {#2} }
2887   }

```

```
2888 }
```

通用注册函数把命令写入 `\g__xeCJK_boundary_registered_prop`。另有一些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 叠加到已有专用适配器外层。

```
2889 \cs_new_protected:Npn \__xeCJK_boundary_reserve:n #1
2890 { \prop_gput:Nnn \g__xeCJK_boundary_reserved_prop {#1} { adapter } }
2891 \cs_new_protected:Npn \__xeCJK_boundary_reserve:N #1
2892 {
2893   \exp_args:Ne \__xeCJK_boundary_reserve:n { \cs_to_str:N #1 }
2894 }
2895 \prg_new_conditional:Npnn \__xeCJK_boundary_if_registered:n #1 { T , F , TF }
2896 {
2897   \prop_if_in:NnTF \g__xeCJK_boundary_registered_prop {#1}
2898   { \prg_return_true: }
2899   {
2900     \prop_if_in:NnTF \g__xeCJK_boundary_reserved_prop {#1}
2901     { \prg_return_true: }
2902     { \prg_return_false: }
2903   }
2904 }
2905 \__xeCJK_msg_new:nn { boundary-register-command }
2906 {
2907   The~value~of~`command'~in~`experiment/boundary-register'~must~be~
2908   exactly~one~control~sequence.
2909 }
2910 \__xeCJK_msg_new:nn { boundary-register-choice }
2911 { Invalid~boundary~registration~choice:~#1. }
2912 \__xeCJK_msg_new:nn { boundary-register-missing }
2913 { The~#1'~entry~is~required~in~`experiment/boundary-register'. }
2914 \__xeCJK_msg_new:nn { boundary-register-mode }
2915 { The~boundary~registration~strategy~and~mode~cannot~be~combined:~#1. }
2916 \__xeCJK_msg_new:nn { boundary-register-duplicate }
2917 { The~command~#1'~has~already~been~declared~for~boundary~registration. }
2918 \__xeCJK_msg_new:nn { boundary-register-conflict }
2919 { The~command~#1'~already~has~an~xeCJK~boundary~registration. }
2920 \__xeCJK_msg_new:nn { boundary-register-late }
2921 {
2922   The~`experiment/boundary-register'~option~can~only~be~used~before~
2923   the~end~of~the~preamble.
2924 }
2925 \__xeCJK_msg_new:nn { boundary-register-undefined }
2926 { The~registered~boundary~command~#1'~is~still~undefined. }
2927
2928 \cs_new_protected:Npn \__xeCJK_boundary_user_register:n #1
2929 {
2930   \bool_if:NTF \g__xeCJK_boundary_user_register_closed_bool
2931   { \__xeCJK_error:n { boundary-register-late } }
2932   {
2933     \bool_set_false:N \l__xeCJK_boundary_user_invalid_bool
2934     \bool_set_false:N \l__xeCJK_boundary_user_mode_given_bool
2935     \tl_clear:N \l__xeCJK_boundary_user_name_tl
2936     \tl_clear:N \l__xeCJK_boundary_user_strategy_tl
2937     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { auto }
2938     \keys_set:nn { xeCJK / boundary-register } {#1}
2939     \tl_if_empty:NT \l__xeCJK_boundary_user_name_tl
2940     {
2941       \__xeCJK_error:ne { boundary-register-missing } { command }
2942       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2943     }
2944     \tl_if_empty:NT \l__xeCJK_boundary_user_strategy_tl
2945     {
2946       \__xeCJK_error:ne { boundary-register-missing } { strategy }
2947       \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2948     }
2949     \bool_if:NF \l__xeCJK_boundary_user_invalid_bool
```

```

2950         { \_xeCJK_boundary_user_register_validate: }
2951         \bool_if:NF \l__xeCJK_boundary_user_invalid_bool
2952         { \_xeCJK_boundary_user_register_store: }
2953     }
2954 }
2955
2956 \cs_new_protected:Npn \_xeCJK_boundary_user_register_command:n #1
2957 {
2958     \tl_if_single_token:nTF {#1}
2959     {
2960         \token_if_cs:NTF #1
2961         { \tl_set:Nx \l__xeCJK_boundary_user_name_tl { \cs_to_str:N #1 } }
2962         {
2963             \_xeCJK_error:n { boundary-register-command }
2964             \tl_set:Nn \l__xeCJK_boundary_user_name_tl { invalid }
2965             \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2966         }
2967     }
2968     {
2969         \_xeCJK_error:n { boundary-register-command }
2970         \tl_set:Nn \l__xeCJK_boundary_user_name_tl { invalid }
2971         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2972     }
2973 }
2974
2975 \cs_new_protected:Npn \_xeCJK_boundary_user_register_strategy:n #1
2976 {
2977     \str_case:nnF {#1}
2978     {
2979         { box } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2980         { wrapped-box } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2981         { stream } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2982         { transparent } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2983         { post-transparent } { \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl {#1} }
2984     }
2985     {
2986         \_xeCJK_error:ne { boundary-register-choice }
2987         { strategy~=\tl_to_str:n {#1} }
2988         \tl_set:Nn \l__xeCJK_boundary_user_strategy_tl { invalid }
2989         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
2990     }
2991 }
2992
2993 \cs_new_protected:Npn \_xeCJK_boundary_user_register_mode:n #1
2994 {
2995     \bool_set_true:N \l__xeCJK_boundary_user_mode_given_bool
2996     \str_case:nnF {#1}
2997     {
2998         { auto } { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
2999         { default } { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
3000         { first-default } { \tl_set:Nn \l__xeCJK_boundary_user_mode_tl {#1} }
3001     }
3002     {
3003         \_xeCJK_error:ne { boundary-register-choice }
3004         { mode~=\tl_to_str:n {#1} }
3005         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3006     }
3007 }
3008
3009 \cs_new_protected:Npn \_xeCJK_boundary_user_register_validate:
3010 {
3011     \str_case:Vn \l__xeCJK_boundary_user_strategy_tl
3012     {
3013         { box }
3014         {
3015             \tl_if_eq:NnT \l__xeCJK_boundary_user_mode_tl { first-default }
3016             {

```

```

3017         \__xeCJK_error:ne { boundary-register-mode } { box/first-default }
3018         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3019     }
3020 }
3021 { wrapped-box }
3022 {
3023     \tl_if_eq:NnT \l__xeCJK_boundary_user_mode_tl { first-default }
3024     {
3025         \__xeCJK_error:ne { boundary-register-mode }
3026         { wrapped-box/first-default }
3027         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3028     }
3029 }
3030 { stream } { }
3031 { transparent }
3032 {
3033     \bool_if:NT \l__xeCJK_boundary_user_mode_given_bool
3034     {
3035         \__xeCJK_error:ne { boundary-register-mode }
3036         { transparent/\l__xeCJK_boundary_user_mode_tl }
3037         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3038     }
3039     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { none }
3040 }
3041 { post-transparent }
3042 {
3043     \bool_if:NT \l__xeCJK_boundary_user_mode_given_bool
3044     {
3045         \__xeCJK_error:ne { boundary-register-mode }
3046         { post-transparent/\l__xeCJK_boundary_user_mode_tl }
3047         \bool_set_true:N \l__xeCJK_boundary_user_invalid_bool
3048     }
3049     \tl_set:Nn \l__xeCJK_boundary_user_mode_tl { none }
3050 }
3051 }
3052 }
3053
3054 \cs_new_protected:Npn \__xeCJK_boundary_user_register_store:
3055 {
3056     \exp_args:NV \__xeCJK_boundary_if_registered:nTF
3057     \l__xeCJK_boundary_user_name_tl
3058     {
3059         \__xeCJK_error:ne { boundary-register-conflict }
3060         { \c_backslash_str \l__xeCJK_boundary_user_name_tl }
3061     }
3062     {
3063         \exp_args:NNV \prop_if_in:NnTF
3064         \g__xeCJK_boundary_user_registered_prop \l__xeCJK_boundary_user_name_tl
3065         {
3066             \__xeCJK_error:ne { boundary-register-duplicate }
3067             { \c_backslash_str \l__xeCJK_boundary_user_name_tl }
3068         }
3069         {
3070             \tl_set:Ne \l__xeCJK_boundary_user_value_tl
3071             {
3072                 \l__xeCJK_boundary_user_strategy_tl
3073                 / \l__xeCJK_boundary_user_mode_tl
3074             }
3075             \exp_args:NNVV \prop_gput:Nnn
3076             \g__xeCJK_boundary_user_registered_prop
3077             \l__xeCJK_boundary_user_name_tl
3078             \l__xeCJK_boundary_user_value_tl
3079         }
3080     }
3081 }
3082
3083 \cs_new_protected:Npn \__xeCJK_boundary_user_register_apply:

```

```

3084 {
3085   \bool_gset_true:N \g__xeCJK_boundary_user_register_closed_bool
3086   \prop_map_inline:Nn \g__xeCJK_boundary_user_registered_prop
3087   {
3088     \__xeCJK_boundary_if_registered:nTF {##1}
3089     {
3090       \__xeCJK_error:ne { boundary-register-conflict }
3091       { \c_backslash_str ##1 }
3092     }
3093     {
3094       \__xeCJK_boundary_user_register_apply:nn {##1} {##2}
3095       \seq_gput_right:Nn \g__xeCJK_boundary_user_applied_seq {##1}
3096     }
3097   }
3098 }
3099
3100 \cs_new_protected:Npn \__xeCJK_boundary_user_register_apply:nn #1#2
3101 {
3102   \seq_set_split:Nnn \l__xeCJK_boundary_user_value_seq { / } {#2}
3103   \seq_pop_left:NN
3104   \l__xeCJK_boundary_user_value_seq \l__xeCJK_boundary_user_strategy_tl
3105   \seq_pop_left:NN
3106   \l__xeCJK_boundary_user_value_seq \l__xeCJK_boundary_user_mode_tl
3107   \str_case:Vn \l__xeCJK_boundary_user_strategy_tl
3108   {
3109     { box }
3110     {
3111       \exp_args:NnV \__xeCJK_boundary_register_box:nn
3112       {#1} \l__xeCJK_boundary_user_mode_tl
3113     }
3114     { wrapped-box }
3115     {
3116       \exp_args:NnV \__xeCJK_boundary_register_wrapped_box:nn
3117       {#1} \l__xeCJK_boundary_user_mode_tl
3118     }
3119     { stream }
3120     {
3121       \exp_args:NnV \__xeCJK_boundary_register_stream:nn
3122       {#1} \l__xeCJK_boundary_user_mode_tl
3123     }
3124     { transparent }
3125     { \__xeCJK_boundary_register_transparent:n {#1} }
3126     { post-transparent }
3127     { \__xeCJK_boundary_register_post_transparent:n {#1} }
3128   }
3129 }
3130
3131 \cs_new_protected:Npn \__xeCJK_boundary_user_register_check:
3132 {
3133   \seq_map_inline:Nn \g__xeCJK_boundary_user_applied_seq
3134   {
3135     \cs_if_exist:cF {##1}
3136     {
3137       \__xeCJK_error:ne { boundary-register-undefined }
3138       { \c_backslash_str ##1 }
3139     }
3140   }
3141 }

```

\mbox 和 \fbox 的公开命令是 robust wrapper; 实际单参数正文由带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 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 \setbox 测量。

```

3142 \cs_new_protected:Npn \__xeCJK_boundary_prepare_onearg_box:N #1
3143 {
3144   \group_begin: \exp_args:NcNc \group_end:

```

```

3145     { __xeCJK_boundary_prepare_onearg_box:NN }
3146     #1 { \cs_to_str:N #1 ~ }
3147   }
3148 \cs_new_protected:Npn \__xeCJK_boundary_prepare_onearg_box:NN #1#2
3149 {
3150   \__xeCJK_boundary_reserve:N #2
3151   \exp_args:Nc \cs_new_eq:NN
3152   { __xeCJK_boundary_orig_\cs_to_str:N #1 _box:n } #2
3153   \cs_gset_protected:Npn #2 ##1
3154   {
3155     \__xeCJK_boundary_math_begin:n {##1}
3156     \use:c { __xeCJK_boundary_orig_\cs_to_str:N #1 _box:n }
3157     { ##1 \__xeCJK_boundary_math_end:n {##1} }
3158   }
3159 }

```

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 transparent capture 的同一 begin/end, 只负责保留零参数或单参数调用签名。

```

3160 \cs_new_protected:Npn \__xeCJK_boundary_wrap_transparent_noarg:NN #1#2
3161 {
3162   \cs_if_exist:NF #1
3163   {
3164     \__xeCJK_boundary_reserve:N #2
3165     \cs_set_eq:NN #1#2
3166     \cs_gset_protected:Npn #2
3167     {
3168       \__xeCJK_boundary_hmode_transparent_begin:
3169       #1
3170       \__xeCJK_boundary_hmode_transparent_end:
3171     }
3172   }
3173 }
3174 \cs_new_protected:Npn \__xeCJK_boundary_wrap_transparent_onearg:NN #1#2
3175 {
3176   \cs_if_exist:NF #1
3177   {
3178     \__xeCJK_boundary_reserve:N #2
3179     \cs_set_eq:NN #1#2
3180     \cs_gset_protected:Npn #2 ##1
3181     {
3182       \__xeCJK_boundary_hmode_transparent_begin:
3183       #1 ##1
3184       \__xeCJK_boundary_hmode_transparent_end:
3185     }
3186   }
3187 }

```

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。

```

3188 \cs_new_protected:Npn \__xeCJK_boundary_wrap_transparent_onearg_braced: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 ##1
3195     {

```

```

3196         \_xeCJK_boundary_hmode_transparent_begin:
3197         #1 {##1}
3198         \_xeCJK_boundary_hmode_transparent_end:
3199     }
3200 }
3201 }

```

这里包装的是本包最低支持内核 $\text{L}\text{A}\text{T}\text{E}\text{X} 2_{\epsilon}$ 2026-06-01 中, 内部 $\backslash\text{imakebox}/\backslash\text{iframbox}$ 的 $[\text{#1}][\text{#2}]\text{#3}$ 参数签名, 不是稳定公共接口。若内核改变该签名, 必须同步 `wrapper`; `command-boundary01` 的 optional $\backslash\text{makebox}/\backslash\text{framebox}$ 场景作为漂移门禁。

```

3202 \cs_new_protected:Npn \_xeCJK_boundary_register_makeboxes:
3203 {
3204     \clist_map_inline:nn { makebox , framebox , @imakebox , @iframbox }
3205     { \_xeCJK_boundary_reserve:n {##1} }
3206     \cs_if_exist:NF \_xeCJK_orig_imakebox:w
3207     {
3208         \cs_new_eq:NN \_xeCJK_orig_imakebox:w \imakebox
3209         \cs_gset_protected:Npn \imakebox [##1][##2]##3
3210         {
3211             \_xeCJK_boundary_inline_last_box_begin:
3212             \_xeCJK_boundary_math_begin:n {##3}
3213             \_xeCJK_orig_imakebox:w [##1][##2]
3214             { ##3 \_xeCJK_boundary_math_end:n {##3} }
3215             \_xeCJK_boundary_inline_last_box_end:n { auto }
3216         }
3217     }
3218     \cs_if_exist:NF \_xeCJK_orig_iframbox:w
3219     {
3220         \cs_new_eq:NN \_xeCJK_orig_iframbox:w \iframbox
3221         \cs_gset_protected:Npn \iframbox [##1][##2]##3
3222         {
3223             \_xeCJK_boundary_inline_last_box_begin:
3224             \_xeCJK_boundary_math_begin:n {##3}
3225             \_xeCJK_orig_iframbox:w [##1][##2]
3226             { ##3 \_xeCJK_boundary_math_end:n {##3} }
3227             \_xeCJK_boundary_inline_last_box_end:n { auto }
3228         }
3229     }
3230 }

```

$\backslash\text{sbox}$ 不能使用通用 `cmd/.../before` 钩子。钩子代码插在 $\backslash\text{setbox}$ 之前执行, 而调用方写的 $\backslash\text{global}$ 是等待下一个赋值的前缀: 钩子里任何赋值都会把它消耗掉, 于是 $\backslash\text{global}\backslash\text{sbox}$ 退化为局部赋值, 盒子在分组结束时丢失, 且不报任何错误 (#1029, `algorithm2e` 的 $\backslash\text{caption}$ 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内部, 使前缀始终紧邻 $\backslash\text{setbox}$ 。

$\backslash\text{savebox}$ 的四种形式 (无可选参数、 $[\text{wd}]$ 、 $[\text{wd}][\text{pos}]$, 以及 `picture` 形式 $(x,y)[\text{pos}]$, 后者经 $\backslash\text{@isavepicbox}$ 汇入) 最终都走同一个 $\backslash\text{sbox}\sim$ 入口, 因此排版路径一并覆盖; 但 $\backslash\text{global}\backslash\text{savebox}$ 跨分组本来就不生效 (前缀在 $\backslash\text{savebox}$ 自己的 $\backslash\text{@ifnextchar}$ 前瞻阶段即被消耗, 原版 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 亦然), 不在本次修复范围内。

与 $\backslash\text{color@b@x}$ 、 $\backslash\text{@textcolor}$ 等适配器不同, 这里不保存并委托原定义, 而是整体覆盖 $\backslash\text{sbox}\sim$: 内核 $\backslash\text{sbox}\sim$ 本体就是赋值语句, 委托写法必须把 $\backslash\text{setbox}$ 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之前 也包装了 $\backslash\text{sbox}\sim$, 它的包装会被丢弃 (反序加载正常)。撰写本节时扫描过完整 TEX Live 树, 未发现任何宏包重定义 $\backslash\text{sbox}\sim$ 或对 $\backslash\text{sbox}$ 安装 `cmd/sbox` 钩子, 故接受这一取舍; $\backslash\text{cs_if_eq:cNF}$ 保证本包自身不会重复包装。

```

3231 \cs_new_protected:Npn \_xeCJK_boundary_sbox:Nn #1#2
3232 {
3233     \tex_setbox:D #1 \tex_hbox:D
3234     {
3235         \_xeCJK_boundary_capture_suspend:
3236         \color@setgroup #2 \color@endgroup
3237         \_xeCJK_boundary_capture_resume:
3238     }

```

```

3239 }
3240 \cs_new_protected:Npn \__xeCJK_boundary_prepare_sbox:
3241 {
3242   \cs_if_eq:cNF { sbox~ } \__xeCJK_boundary_sbox:Nn
3243   { \cs_gset_eq:cN { sbox~ } \__xeCJK_boundary_sbox:Nn }
3244 }
3245
3246 \__xeCJK_boundary_prepare_onearg_box:N \mbox
3247 \__xeCJK_boundary_prepare_onearg_box:N \fbox
3248 \__xeCJK_boundary_register_box:nn { mbox } { auto }
3249 \__xeCJK_boundary_register_box:nn { fbox } { auto }
3250 \__xeCJK_boundary_register_makeboxes:
3251 \__xeCJK_boundary_register_post_transparent:n { null }
3252 \__xeCJK_boundary_reserve:n { sbox }
3253 \__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。

```

3254 \cs_new_eq:NN \xeCJK_make_space_node: \prg_do_nothing:
3255 \cs_new_protected:Npe \__xeCJK_make_space_node:
3256 {
3257   \tex_hskip:D - \__xeCJK_glue_node:n { default-space }
3258   \tex_hskip:D \__xeCJK_glue_node:n { default-space }
3259 }

```

CJKglue CJK 文字之间插入的 glue。

```

3260 \keys_define:nn { xeCJK / options }
3261 {
3262   CJKglue .code:n =
3263   {
3264     \cs_set_protected:Npn \CJKglue {#1}
3265     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
3266   }
3267 }
3268 \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

```

3269 \keys_define:nn { xeCJK / options }
3270 {
3271   CJKecglue .code:n =
3272   {
3273     \cs_set_protected:Npn \CJKecglue {#1}
3274     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
3275   } ,
3276   xCJKecglue .choice: ,
3277   xCJKecglue / true .code:n =
3278   {
3279     \bool_set_true:N \l__xeCJK_xecglue_bool
3280     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
3281     \cs_set_eq:NN \xeCJK_make_space_node: \__xeCJK_make_space_node:
3282     \cs_set_eq:NN \xeCJK_check_for_xglue: \__xeCJK_check_for_xglue:
3283     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_xecglue:
3284     \cs_set_eq:NN
3285     \xeCJK_check_for_ecglue_normalsp:
3286     \__xeCJK_check_for_xecglue_normalsp:
3287   } ,
3288   xCJKecglue / false .code:n =
3289   {
3290     \bool_set_false:N \l__xeCJK_xecglue_bool
3291     \cs_set_eq:NN \xeCJK_space_or_xecglue: \xeCJK_space_glue:
3292     \xeCJK_cs_clear:N \xeCJK_make_space_node:
3293     \xeCJK_cs_clear:N \xeCJK_check_for_xglue:
3294     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:

```

```

3295         \cs_set_eq:NN
3296         \xeCJK_check_for_ecglue_normalsp:
3297         \__xeCJK_check_for_ecglue_normalsp:
3298     } ,
3299     xCJKecglue / unknown .code:n =
3300     {
3301         \bool_set_true:N \l__xeCJK_xecglue_bool
3302         \cs_set_protected:Npn \CJKecglue {#1}
3303         \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
3304         \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
3305         \cs_set_eq:NN \xeCJK_make_space_node: \__xeCJK_make_space_node:
3306         \cs_set_eq:NN \xeCJK_check_for_xglue: \__xeCJK_check_for_xglue:
3307         \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_xecglue:
3308         \cs_set_eq:NN
3309         \xeCJK_check_for_ecglue_normalsp:
3310         \__xeCJK_check_for_xecglue_normalsp:
3311     } ,
3312     xCJKecglue .default:n = { true }
3313 }
3314 \cs_new_eq:NN \xeCJK_space_glue: \c_space_tl
3315 \skip_new:N \l__xeCJK_ecglue_skip
3316 \bool_new:N \l__xeCJK_xecglue_bool

```

CJKspace 是否保留 CJK 文字间的空白, 默认不保留。

```

3317 \keys_define:nn { xeCJK / options }
3318 {
3319     CJKspace .choice: ,
3320     CJKspace / true .code:n =
3321     {
3322         \bool_set_true:N \l__xeCJK_reserve_space_bool
3323         \cs_set_protected:Npn \__xeCJK_ccglue_or_space:
3324         { \xeCJK_space_glue: }
3325     } ,
3326     CJKspace / false .code:n =
3327     {
3328         \bool_set_false:N \l__xeCJK_reserve_space_bool
3329         \cs_set_protected:Npn \__xeCJK_ccglue_or_space:
3330         { \CJKglue }
3331     } ,
3332     CJKspace .default:n = { true } ,
3333     space .meta:n = { CJKspace = true } ,
3334     nospace .meta:n = { CJKspace = false }
3335 }
3336 \bool_new:N \l__xeCJK_reserve_space_bool
3337 \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。

```

3338 \cs_new_protected:Npn \xeCJK_CJK_and_Boundary:w
3339 {
3340     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
3341     { \__xeCJK_CJK_and_math: }
3342     {
3343         \group_align_safe_begin:
3344         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3345         {
3346             \token_if_macro:NTF \l_peek_token
3347             {
3348                 \token_if_eq_meaning:NNTF
3349                 \l_peek_token \__xeCJK_boundary_identity:n
3350                 { \__xeCJK_boundary_peek_math:w }

```

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

```

3351         { \_xeCJK_boundary_reserve_space: }
3352     }
3353     {
3354         \token_if_group_begin:NTF \l_peek_token
3355         { \_xeCJK_boundary_group_math:w }
3356         { \_xeCJK_boundary_group_end:n { CJK-space } }
3357     }
3358 }
3359 {
3360     \token_if_eq_meaning:NNTF \l_peek_token \scan_stop:
3361     { \_xeCJK_CJK_and_Boundary_relax:N }
3362     {
3363         \token_if_group_begin:NTF \l_peek_token
3364         { \_xeCJK_boundary_group_math:w }
3365         {
3366             \token_if_group_end:NTF \l_peek_token
3367             {
3368                 \bool_gset_true:N \g_xeCJK_glue_check_pending_bool
3369                 \_xeCJK_boundary_group_end:n { CJK }
3370             }
3371             {
3372                 \token_if_macro:NTF \l_peek_token
3373                 {
3374                     \token_if_eq_meaning:NNTF
3375                     \l_peek_token \_xeCJK_boundary_identity:n
3376                     { \_xeCJK_boundary_peek_math:w }
3377                     { \_xeCJK_boundary_group_end:n { CJK } }
3378                 }
3379                 { \_xeCJK_boundary_group_end:n { CJK } }
3380             }
3381         }
3382     }
3383 }
3384 }
3385 }
3386 \cs_new_protected:Npn \_xeCJK_CJK_and_math:
3387 {
3388     \bool_if:NTF \l_xeCJK_peek_ignore_spaces_bool
3389     { \xeCJK_class_group_end: \xeCJK_space_or_xecglue: }
3390     { \xeCJK_class_group_end: \CJKecglue }
3391 }
3392 \cs_new_protected:Npn \_xeCJK_boundary_CJK_and_math:
3393 { \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。

```

3394 \cs_new_protected:Npn \_xeCJK_boundary_group_math:w
3395 {
3396     \tex_afterassignment:D \_xeCJK_boundary_group_math_peek:
3397     \tex_let:D \_xeCJK_boundary_group_math_brace:
3398 }
3399 \cs_new_eq:NN \_xeCJK_boundary_group_math_brace: \scan_stop:

```

```

3400 \cs_new_protected:Npn \__xeCJK_boundary_group_math_peek:
3401 { \peek_after:Nw \__xeCJK_boundary_group_math_branches: }
3402 \cs_new_protected:Npn \__xeCJK_boundary_group_math_branches:
3403 {
3404   \token_if_math_toggle:NTF \l_peek_token
3405   { \__xeCJK_boundary_CJK_and_math: }
3406   {
3407     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3408     { \__xeCJK_boundary_reserve_space: }
3409     { \__xeCJK_boundary_group_end:n { CJK } }
3410   }
3411   \c_group_begin_token
3412 }

```

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。

```

3413 \cs_new:Npn \__xeCJK_boundary_identity:n #1 {#1}
3414 \cs_new_protected:Npn \__xeCJK_boundary_peek_math:w
3415 {
3416   \exp_after:wN \peek_after:Nw
3417   \exp_after:wN \__xeCJK_boundary_peek_math_branches:w
3418   \exp:w \exp_end_continue_f:w
3419 }
3420 \cs_new_protected:Npn \__xeCJK_boundary_peek_math_branches:w
3421 {
3422   \token_if_math_toggle:NTF \l_peek_token
3423   { \__xeCJK_boundary_CJK_and_math: }
3424   {
3425     \token_if_group_begin:NTF \l_peek_token
3426     { \__xeCJK_boundary_group_math:w }
3427     {
3428       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3429       { \__xeCJK_boundary_reserve_space: }
3430       { \__xeCJK_boundary_group_end:n { CJK } }
3431     }
3432   }
3433 }
3434 \cs_new_protected:Npn \__xeCJK_boundary_reserve_space:
3435 { \__xeCJK_boundary_group_end:n { CJK-space } }
3436 \cs_new_protected:Npn \__xeCJK_CJK_and_Boundary_relax:N #1
3437 {
3438   \__xeCJK_boundary_group_end:n { CJK }
3439   \token_if_eq_meaning:NNTF #1 \scan_stop:
3440   {#1} { \cs_set_eq:NN #1 \scan_stop: #1 }
3441 }
3442 \cs_new_protected:Npn \__xeCJK_boundary_group_end:n #1
3443 {
3444   \group_align_safe_end:
3445   \xeCJK_class_group_end:
3446   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
3447   { \xeCJK_make_node:n {#1} }
3448 }
\xeCJK_ignore_spaces:w 3449 \cs_new_protected:Npn \xeCJK_ignore_spaces:w
3450 {
3451   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
3452   {
3453     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
3454     { \xeCJK_space_or_xecglue: } { \CJKecglue }
3455   }
3456   {
3457     \bool_if:NT \l__xeCJK_peek_ignore_spaces_bool
3458     {
3459       \dim_case:nn { \tex_lastkern:D }
3460       {
3461         { \__xeCJK_node:n { CJK } }
3462         { \xeCJK_remove_node: \xeCJK_make_node:n { CJK-space } }

```

```

3463         { \_xeCJK_node:n { default } }
3464         { \xeCJK_remove_node: \xeCJK_make_space_node: }
3465     }
3466     \group_align_safe_begin:
3467     \token_if_macro:NTF \l_peek_token
3468     { \_xeCJK_reserve_space_aux: }
3469     { \group_align_safe_end: }
3470 }
3471 }
3472 }
3473 \cs_new_protected:Npn \_xeCJK_reserve_space_aux:
3474 {
3475     \group_align_safe_end:
3476     \xeCJK_space_or_xecglue:
3477 }
3478 \xeCJK_inter_class_toks:nnn { CJK } { CJK }
3479 { \xeCJK_CJK_and_CJK:N }
\xeCJK_CJK_and_CJK:N 3480 \cs_new_protected:Npn \xeCJK_CJK_and_CJK:N
3481 {
3482     \CJKglue
3483     \xeCJK_fallback_symbol:NN
3484     \CJKsymbol
3485 }
3486 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
3487 {
3488     \xeCJK_FullLeft_and_CJK:
3489     \xeCJK_fallback_symbol:NN
3490     \CJKsymbol
3491 }
3492 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
3493 {
3494     \xeCJK_FullRight_and_CJK:
3495     \xeCJK_fallback_symbol:NN
3496     \CJKsymbol
3497 }
3498 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
3499 {
3500     \clist_map_inline:nn { FullLeft , FullRight }
3501     {
3502         \xeCJK_inter_class_toks:nne {#1} {##1}
3503         { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
3504         \xeCJK_inter_class_toks:nne {##1} {#1}
3505         { \exp_not:c { xeCJK_##1_and_Default: } }
3506     }
3507 }
\l__xeCJK_halfright_nobreak_bool 3508 \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。

```

3509 \xeCJK_app_inter_class_toks:nnn { CJK } { HalfRight }
3510 {
3511     \bool_if:NT \l__xeCJK_halfright_nobreak_bool
3512     { \xeCJK_no_break: }
3513 }

```

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

```

3514 \xeCJK_inter_class_toks:nnn { FullRight } { HalfRight }
3515 {
3516     \_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3517     \xeCJK_class_group_end:
3518     \bool_if:NT \l__xeCJK_halfright_nobreak_bool
3519     { \xeCJK_no_break: }
3520     \_xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3521 }

```

```

3522 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
3523 { \xeCJK_Boundary_and_FullLeft:N }
3524 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
3525 { \xeCJK_Boundary_and_FullRight:N }

3526 \xeCJK_inter_class_toks:nnn { FullLeft } { Boundary }
3527 { \xeCJK_FullLeft_and_Boundary: }
3528 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
3529 { \xeCJK_FullRight_and_Boundary: }

\xeCJK_FullLeft_and_Boundary: 3530 \cs_new_protected:Npn \xeCJK_FullLeft_and_Boundary:
3531 {
3532   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3533   {
3534     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3535     \xeCJK_class_group_end:
3536     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3537     \xeCJK_no_break:
3538     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3539     \__xeCJK_punct_boundary_guard:
3540   }
3541   {
3542     \xeCJK_class_group_end:
3543     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3544     \__xeCJK_nobreak_zero_glue:
3545   }
3546   \tex_ignorespaces:D
3547 }

```

```

\xeCJK_FullRight_and_Boundary: 3548 \cs_new_protected:Npn \xeCJK_FullRight_and_Boundary:
3549 {
3550   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3551   \xeCJK_class_group_end:
3552   \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
3553   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3554   \__xeCJK_punct_boundary_guard:
3555   \tex_ignorespaces:D
3556 }

```

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

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

```

3557 \bool_new:N \g__xeCJK_par_guard_bool
3558 \dim_new:N \g__xeCJK_par_guard_dim
3559 \cs_new_protected:Npn \__xeCJK_punct_boundary_guard:
3560 {
3561   \mode_if_inner:TF
3562   { \tex_penalty:D \c_zero_int }
3563   {
3564     \bool_if:NT \g__xeCJK_punct_measure_fix_bool
3565     {
3566       \bool_gset_true:N \g__xeCJK_par_guard_bool
3567       \dim_gset:Nn \g__xeCJK_par_guard_dim { \tex_lastskip:D }
3568     }
3569   }
3570 }

```

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。

```

3571 \clist_map_inline:nn
3572 {
3573   Default , HalfLeft , HalfRight , NormalSpace ,
3574   CJK , FullLeft , FullRight , CM , HangulJamo ,

```

```

3575     HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
3576   }
3577   {
3578     \xeCJK_pre_inter_class_toks:nnn { Boundary } {#1}
3579     { \bool_gset_false:N \g__xeCJK_par_guard_bool }
3580   }

```

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。

```

3581 \AddToHook { para/begin }
3582 { \bool_gset_false:N \g__xeCJK_par_guard_bool }
3583 \AddToHook { para/end }
3584 {
3585   \bool_if:NT \g__xeCJK_par_guard_bool
3586   {
3587     \tex_kern:D \g__xeCJK_par_guard_dim
3588     \bool_gset_false:N \g__xeCJK_par_guard_bool
3589   }
3590 }

```

\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。

```

3591 \cs_new_protected:Npn \xeCJK_punct_node:N #1
3592 {
3593   \__xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
3594   \__xeCJK_make_node:N \l__xeCJK_tmp_dim
3595   \dim_set:Nn \l__xeCJK_tmp_dim { `#1 sp }
3596   \__xeCJK_make_node:N \l__xeCJK_tmp_dim
3597 }

```

__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”。

```

3598 \cs_new_protected:Npn \__xeCJK_punct_bound_unitization:NN #1#2
3599 {
3600   \dim_set:Nn #2
3601   {
3602     \dim_max:nn
3603     { \c_zero_dim }
3604     { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl #1 }
3605   }
3606   \dim_compare:nNnF {#2} < { 1pt }
3607   { \dim_set:Nn #2 { -1pt * \dim_ratio:nn {#2} { \c_max_dim } } }
3608 }

```

```

\xeCJK_punct_bound_kern:N 3609 \cs_new_protected:Npn \xeCJK_punct_bound_kern:N #1
__xeCJK_punct_bound_kern:NN 3610 {
3611   \exp_after:wN \__xeCJK_punct_bound_kern:NN
3612   \g__xeCJK_last_punct_tl #1
3613 }
3614 \cs_new_protected:Npn \__xeCJK_punct_bound_kern:NN #1#2
3615 {
3616   \xeCJK_get_punct_bounds:NN \l__xeCJK_aligni_tl #1
3617   \xeCJK_get_punct_kerning:NN #1 #2
3618   \__xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
3619   \skip_set:Nn \l__xeCJK_punct_kern_skip
3620   { \__xeCJK_use_dim_or_skip:nnn { bound_kern } #1 #2 }
3621   \dim_compare:nNnF \l__xeCJK_tmp_dim = \l__xeCJK_last_bound_dim
3622   { \__xeCJK_punct_bound_kern_ratio:NN #1 #2 }
3623   \bool_if:NTF \l__xeCJK_last_penalty_bool
3624   {
3625     \tex_penalty:D \l__xeCJK_last_penalty_int
3626     \skip_horizontal:N
3627   }
3628   { \__xeCJK_punct_bound_kern_aux:NNN #1 #2 }
3629   \l__xeCJK_punct_kern_skip
3630 }
3631 \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。

```

3632 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_ratio:NN #1#2
3633 {
3634   \dim_set:Nn \l__xeCJK_bound_dim
3635     { \_xeCJK_use_punct_dim:nNN { bound_width } #1 #2 }
3636   \dim_compare:nNnT \l__xeCJK_bound_dim > \c_zero_dim
3637     {
3638       \dim_compare:nNnF \l__xeCJK_last_bound_dim > \c_zero_dim
3639       {
3640         \dim_set:Nn \l__xeCJK_last_bound_dim
3641         {
3642           - \l__xeCJK_last_bound_dim *
3643             \dim_ratio:nn { \c_max_dim } { 1pt }
3644         }
3645       }
3646       \_xeCJK_punct_bound_kern_ratio_aux:N #2
3647     }
3648 }
3649 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_ratio_aux:N #1
3650 {
3651   \skip_set:Nn \l__xeCJK_punct_kern_skip
3652   {
3653     \l__xeCJK_punct_kern_skip *
3654     \dim_ratio:nn
3655     {
3656       \l__xeCJK_last_bound_dim
3657       + \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1
3658     }
3659     { \l__xeCJK_bound_dim }
3660   }
3661 }

\_xeCJK_nobreak_hskip:N 3662 \cs_new_protected:Npn \_xeCJK_nobreak_hskip:N
\_xeCJK_nobreak_hskip:n 3663 { \xeCJK_no_break: \skip_horizontal:N }
\_xeCJK_punct_bound_kern:N 3664 \cs_new_protected:Npn \_xeCJK_nobreak_hskip:n
\_xeCJK_punct_bound_breakable_kern:N 3665 { \xeCJK_no_break: \skip_horizontal:n }
3666 \cs_new_eq:NN \_xeCJK_punct_bound_kern:N \_xeCJK_nobreak_hskip:N
3667 \cs_new_protected:Npn \_xeCJK_punct_bound_breakable_kern:N
3668 {
3669   \tl_if_eq:NNTF \l__xeCJK_aligni_tl \c__xeCJK_right_tl
3670   {
3671     \tl_if_eq:NNTF \l__xeCJK_alignii_tl \c__xeCJK_left_tl
3672     { \skip_horizontal:N }
3673     { \_xeCJK_nobreak_hskip:N }
3674   }
3675   { \_xeCJK_nobreak_hskip:N }
3676 }
3677 \cs_new_protected:Npn \_xeCJK_punct_bound_kern_aux:NNN #1#2
3678 {
3679   \str_if_eq:nnTF {#1} {#2}
3680   { \_xeCJK_nobreak_hskip:N }
3681   {
3682     \_xeCJK_punct_if_long:NNTF #1
3683     { \skip_horizontal:N }
3684     {
3685       \_xeCJK_punct_if_long:NNTF #2
3686       { \skip_horizontal:N }
3687       { \_xeCJK_punct_bound_kern:N }
3688     }
3689   }
3690 }

3691 \clist_map_inline:nn { CJK , FullLeft , FullRight }
3692 {
3693   \clist_map_inline:nn { FullLeft , FullRight }
3694   {
3695     \xeCJK_inter_class_toks:nne {#1} {##1}

```

```

3696      { \exp_not:c { xeCJK_#1_and_##1:N } }
3697    }
3698  }

```

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

```

3699 \cs_new_protected:Npn \_xeCJK_punct_bound_rule:NN #1#2
3700 {
3701   \tex_vrule:D
3702     width - \_xeCJK_use_punct_dim:nNN { bound } #1 #2 ~
3703     depth \c_zero_dim
3704     height \c_zero_dim \scan_stop:
3705 }

```

_xeCJK_punct_rule:NN 用于减少标点符号的左/右空白。

```

3706 \cs_new_protected:Npn \_xeCJK_punct_rule:NN #1#2
3707 {
3708   \tex_vrule:D
3709     width \_xeCJK_use_punct_dim:nNN { rule } #1 #2 ~
3710     depth \c_zero_dim
3711     height \c_zero_dim \scan_stop:
3712 }

```

_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。

```

3713 \cs_new_protected:Npn \_xeCJK_punct_glue:NN #1#2
3714 { \_xeCJK_punct_hskip:n { \_xeCJK_use_dim_or_skip:nNN { glue } #1 #2 } }
3715 \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。

```

3716 \cs_new_protected:Npn \xeCJK_punct_kern:NN #1#2
3717 {
3718   \str_if_eq:eeTF {#1} {#2}
3719   { \_xeCJK_punct_nobreak_kern:NN }
3720   {
3721     \bool_lazy_or:nnTF
3722       { \_xeCJK_punct_if_long_p:N #1 }
3723       { \_xeCJK_punct_if_long_p:N #2 }
3724       { \_xeCJK_punct_kern_break:NN #1 #2 }
3725       { \_xeCJK_punct_nobreak_kern:NN }
3726   }
3727   #1 #2
3728 }
3729 \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。

```

3730 \cs_new_protected:Npn \_xeCJK_punct_kern_break:NN #1#2
3731 {
3732   \exp_after:wN \_xeCJK_punct_if_right:NTF #1
3733   {
3734     \_xeCJK_punct_if_right:NTF #2
3735     {
3736       \bool_lazy_and:nnTF
3737         { \_xeCJK_punct_if_long_p:N #2 }
3738         { ! \_xeCJK_punct_if_nobreak_long_p:N #2 }
3739         { \_xeCJK_punct_breakable_kern:NN }

```

```

3740          { \__xeCJK_punct_nobreak_kern:NN }
3741        }
3742        { \__xeCJK_punct_breakable_kern:NN }
3743      }
3744      { \__xeCJK_punct_nobreak_kern:NN }
3745    }

```

```

\__xeCJK_punct_nobreak_kern:NN 3746 \cs_new_protected:Npn \__xeCJK_punct_nobreak_kern:NN #1#2
3747 { \__xeCJK_nobreak_hskip:n { \__xeCJK_use_dim_or_skip:nNN { kern } #1 #2 } }

```

```

\__xeCJK_punct_breakable_kern:NN 3748 \cs_new_protected:Npn \__xeCJK_punct_breakable_kern:NN #1#2
3749 {
3750   \exp_after:wN \__xeCJK_punct_if_right:NT #1
3751   { \__xeCJK_punct_rule:NN \c__xeCJK_right_tl #1 }
3752   \__xeCJK_punct_breakable_kern:n
3753   { \__xeCJK_use_dim_or_skip:nNN { bound_kern } #1 #2 }
3754   \__xeCJK_punct_if_right:NF #2
3755   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #2 }
3756 }
3757 \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。

```

3758 \tl_new:N \g__xeCJK_last_punct_tl

```

```

\xeCJK_FullLeft_and_CJK: 3759 \cs_new_protected:Npn \xeCJK_FullLeft_and_CJK:
3760 {
3761   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3762   {
3763     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3764     \xeCJK_no_break:
3765     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3766   }
3767   { }
3768   \__xeCJK_select_font:
3769 }

```

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

```

3770 \cs_new_protected:Npn \xeCJK_FullLeft_and_Default:
3771 {
3772   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
3773   {
3774     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3775     \xeCJK_class_group_end: \xeCJK_no_break:
3776     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
3777   }
3778   {
3779     \xeCJK_class_group_end:
3780     \__xeCJK_nobreak_zero_glue:
3781   }
3782 }
3783 \cs_new_protected:Npn \__xeCJK_nobreak_zero_glue:
3784 {
3785   \tex_penalty:D \c__xeCJK_nobreak_penalty_int
3786   \skip_horizontal:N \c_zero_skip
3787 }
3788 \cs_new_protected:Npn \__xeCJK_zero_glue:
3789 { \skip_horizontal:N \c_zero_skip }

```

```

\xeCJK_FullRight_and_CJK: 3790 \cs_new_protected:Npn \xeCJK_FullRight_and_CJK:
3791 {
3792   \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3793   \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
3794   \__xeCJK_select_font:
3795   \CJKglue
3796 }

```

```

\XeCJK_FullRight_and_Default: 3797 \cs_new_protected:Npn \XeCJK_FullRight_and_Default:
3798 {
3799   \__XeCJK_punct_rule:NN \c__XeCJK_right_tl \g__XeCJK_last_punct_tl
3800   \XeCJK_class_group_end:
3801   \__XeCJK_punct_glue:NN \c__XeCJK_right_tl \g__XeCJK_last_punct_tl
3802 }

\XeCJK_Default_and_FullLeft:nN 3803 \cs_new_protected:Npn \XeCJK_Default_and_FullLeft:nN #1#2
3804 {
3805   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #2
3806   \__XeCJK_Default_and_FullLeft_glue:N #2
3807   \XeCJK_class_group_begin:
3808   \XeCJK_select_punct_font:
3809   \XeCJK_clear_inter_class_toks:nn {#1} { FullLeft }
3810   \XeCJK_clear_Boundary_and_CJK_toks:
3811   \tl_gset:Nn \g__XeCJK_last_punct_tl {#2}
3812   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #2
3813   \XeCJK_fallback_punct_symbol:NN
3814   \CJKpunctsymbol #2
3815 }
3816 \cs_new_protected:Npn \__XeCJK_Default_and_FullLeft_glue:N #1
3817 { \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1 }

\XeCJK_CJK_and_FullLeft:N 3818 \cs_new_protected:Npn \XeCJK_CJK_and_FullLeft:N #1
3819 {
3820   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #1
3821   \__XeCJK_CJK_and_FullLeft_glue:N #1
3822   \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
3823   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #1
3824   \__XeCJK_select_punct_font:
3825   \XeCJK_fallback_punct_symbol:NN
3826   \CJKpunctsymbol #1
3827 }
3828 \cs_new_protected:Npn \__XeCJK_CJK_and_FullLeft_glue:N #1
3829 {
3830   \CJKglue
3831   \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1
3832 }

\XeCJK_Boundary_and_FullLeft:N 3833 \cs_new_protected:Npn \XeCJK_Boundary_and_FullLeft:N #1
3834 {
3835   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #1
3836   \__XeCJK_Boundary_and_FullLeft_glue:N #1
3837   \XeCJK_class_group_begin:
3838   \XeCJK_select_punct_font:
3839   \XeCJK_clear_Boundary_and_CJK_toks:
3840   \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
3841   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #1
3842   \XeCJK_fallback_punct_symbol:NN
3843   \CJKpunctsymbol #1
3844 }

```

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

```

3845 \cs_new_protected:Npn \__XeCJK_Boundary_and_FullLeft_glue:N #1
3846 {
3847   \tl_set_eq:NN \l__XeCJK_alignii_tl \c__XeCJK_left_tl
3848   \group_begin: \exp_args:NNc \group_end: \cs_if_exist_use:NTF
3849   { \__XeCJK_bound_type_ \int_use:N \etex_lastnodetype:D _glue:Nn }
3850   {#1}
3851   { \use:n }
3852   { \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1 }
3853 }
3854 \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。

```

3855 \cs_new_protected:cpn { \__XeCJK_bound_type_-1 _glue:Nn } #1#2

```

```
3856 { \_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, $\text{\TeX}$ 会在段落开头插入宽度为 `\parindent` 的水平盒子用于缩进。

```
3857 \cs_new_protected:cpn { \_xeCJK_bound_type_ 1 _glue:Nn } #1
3858 {
3859   \int_do_while:nNnn \tex_lastnodetype:D = \c_xeCJK_hlist_node
3860   { \_xeCJK_bound_hbox_auxi: }
3861   \_xeCJK_if_last_none:TF
3862   {
3863     \dim_case:nnF { \box_wd:N \l_xeCJK_indent_box }
3864     {
3865       { \tex_parindent:D } { \_xeCJK_bound_hbox_auxii:nn }
3866       { \c_zero_dim } { \use_i:nn }
3867     }
3868     { \use:nn }
3869   }
3870   { \use:nn }
3871   { \hbox_unpack_drop:N \l_xeCJK_indent_box }
3872 }
3873 \cs_new_protected:Npn \_xeCJK_bound_hbox_auxi:
3874 {
3875   \box_set_to_last:N \l_xeCJK_tmp_box
3876   \hbox_set:Nn \l_xeCJK_indent_box
3877   {
3878     \box_use:N \l_xeCJK_tmp_box
3879     \hbox_unpack:N \l_xeCJK_indent_box
3880   }
3881 }
3882 \cs_new_protected:Npn \_xeCJK_bound_hbox_auxii:nn
3883 {
3884   \dim_compare:nNnTF
3885   { \box_ht:N \l_xeCJK_tmp_box } = \c_zero_dim
3886   { \use_i:nn }
3887   { \use:nn }
3888 }
3889 \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 $\text{\LaTeX}$ 表格的非 p 列行首时, 能够对齐到单元格的边界。判断基于标准 $\text{\LaTeX}$ 表格的列格式 (`\@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`。

```
3890 \cs_new_protected:cpn { \_xeCJK_bound_type_ 11 _glue:Nn } #1#2
3891 {
3892   \skip_if_finite:nTF { \tex_lastskip:D }
3893   { \_xeCJK_bound_glue_auxi:Nn #1 {#2} }
3894   { \_xeCJK_zero_glue: }
3895 }
3896 \cs_new_protected:Npn \_xeCJK_bound_glue_auxi:Nn #1#2
3897 {
3898   \_xeCJK_if_last_punct_glue:TF
3899   { \xeCJK_punct_bound_kern:N #1 }
3900   { \_xeCJK_bound_glue_auxii:n {#2} }
3901 }
3902 \cs_new_protected:Npn \_xeCJK_bound_glue_auxii:n #1
3903 {
3904   \skip_set_eq:NN \l_xeCJK_last_skip \tex_lastskip:D
3905   \skip_if_eq:nnTF { \l_xeCJK_last_skip } { 1sp }
3906   { \_xeCJK_zero_glue: }
3907   {
3908     \skip_if_eq:nnTF { \l_xeCJK_last_skip } { \labelsep }
3909     {
3910       \tex_unskip:D

```

```

3911         \_xeCJK_if_last_penalty:TF
3912         {
3913             \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
3914             { \skip_horizontal:N \l__xeCJK_last_skip }
3915             { \skip_horizontal:N \l__xeCJK_last_skip #1 }
3916         }
3917         { \skip_horizontal:N \l__xeCJK_last_skip #1 }
3918     }
3919     {#1}
3920 }
3921 }

```

`\_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`。

```

3922 \cs_new_protected:cpn { \_xeCJK_bound_type_12_glue:Nn } #1#2
3923 {
3924     \xeCJK_if_last_node:nF { CJK }
3925     { \xeCJK_if_last_node:nF { CJK-space } { \use_none:nn } }
3926     \xeCJK_remove_node: \CJKglue
3927     #2
3928 }

```

`\_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。

```

3929 \cs_new_protected:cpn { \_xeCJK_bound_type_13_glue:Nn } #1#2
3930 {
3931     \_xeCJK_if_last_punct_penalty:TF
3932     { \xeCJK_punct_bound_kern:N #1 }
3933     {
3934         \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
3935         {
3936             \tex_unpenalty:D
3937             \_xeCJK_if_last_hlist:TF
3938             { \tex_penalty:D \c_zero_int }
3939             { \tex_penalty:D \c_zero_int #2 }
3940         }
3941         {#2}
3942     }
3943 }

```

```

\xeCJK_Default_and_FullRight:nN 3944 \cs_new_protected:Npn \xeCJK_Default_and_FullRight:nN #1#2
3945 {
3946     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #2
3947     \_xeCJK_Default_and_FullRight_glue:N #2
3948     \xeCJK_class_group_begin:
3949     \xeCJK_select_punct_font:
3950     \xeCJK_clear_inter_class_toks:nn {#1} { FullRight }
3951     \xeCJK_clear_Boundary_and_CJK_toks:
3952     \tl_gset:Nn \g__xeCJK_last_punct_tl {#2}
3953     \xeCJK_FullRight_symbol:N #2
3954 }

```

```

\xeCJK_Boundary_and_FullRight:N 3955 \cs_new_protected:Npn \xeCJK_Boundary_and_FullRight:N #1
3956 {
3957     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
3958     \xeCJK_if_last_punct:TF
3959     {
3960         \tl_set_eq:NN \l__xeCJK_alignii_tl \c__xeCJK_right_tl
3961         \xeCJK_punct_bound_kern:N
3962     }
3963     { \_xeCJK_Default_and_FullRight_glue:N }
3964     #1
3965     \xeCJK_class_group_begin:
3966     \xeCJK_select_punct_font:
3967     \xeCJK_clear_Boundary_and_CJK_toks:
3968     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}

```

```

3969 \xeCJK_FullRight_symbol:N #1
3970 }

```

```

\xeCJK_CJK_and_FullRight:N 3971 \cs_new_protected:Npn \xeCJK_CJK_and_FullRight:N #1
3972 {
3973   \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
3974   \__xeCJK_CJK_and_FullRight_glue:N #1
3975   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
3976   \__xeCJK_select_punct_font:
3977   \xeCJK_FullRight_symbol:N #1
3978 }

```

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

```

3979 \cs_new_protected:Npn \xeCJK_if_last_punct:TF
3980 {
3981   \bool_set_false:N \l__xeCJK_last_penalty_bool
3982   \__xeCJK_if_last_glue:TF
3983   { \__xeCJK_if_last_punct_glue:TF }
3984   {
3985     \__xeCJK_if_last_penalty:TF
3986     { \__xeCJK_if_last_punct_penalty:TF }
3987     { \use_ii:nn }
3988   }
3989 }
3990 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue:TF
3991 {
3992   \prop_get:NpNTF \g__xeCJK_punct_skip_prop
3993   { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
3994   { \__xeCJK_if_last_punct_glue_auxi:TF }
3995   { \__xeCJK_if_last_punct_glue_auxii:TF }
3996 }
3997 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxi:TF
3998 {
3999   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
4000   \tex_unskip:D
4001   \int_compare:nNnTF \tex_lastpenalty:D = \c__xeCJK_nobreak_penalty_int
4002   { \__xeCJK_if_last_punct_auxi:TF { \use_i:nn } }
4003   {
4004     \xeCJK_if_last_node:TF
4005     { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
4006     { \use:n }
4007   }
4008   { \skip_horizontal:N \l__xeCJK_last_skip \use_ii:nn }
4009 }
4010 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxii:TF
4011 {
4012   \group_begin:
4013   \g__xeCJK_space_factor_int \tex_spacefactor:D
4014   \skip_if_eq:nnTF { \tex_lastskip:D } { \c_xeCJK_space_skip_tl }
4015   { \group_end: \__xeCJK_if_last_punct_glue_auxiii:TF }
4016   { \group_end: \use_ii:nn }
4017 }
4018 \cs_new_protected:Npn \__xeCJK_if_last_punct_glue_auxiii:TF
4019 {
4020   \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
4021   \tex_unskip:D
4022   \__xeCJK_if_last_glue:TF
4023   {
4024     \prop_get:NpNTF \g__xeCJK_punct_skip_prop
4025     { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
4026     { \__xeCJK_if_last_punct_glue_auxi:TF { \use_i:nn } }
4027     { \use:n }
4028   }
4029   { \use:n }
4030   { \skip_horizontal:N \l__xeCJK_tmp_skip \use_ii:nn }
4031 }

```

```

4032 \cs_new_protected:Npn \__xeCJK_if_last_punct_penalty:TF
4033 {
4034   \int_set_eq:NN \l__xeCJK_last_penalty_int \tex_lastpenalty:D
4035   \tex_unpenalty:D
4036   \bool_set_true:N \l__xeCJK_last_penalty_bool
4037   \__xeCJK_if_last_glue:TF
4038   { \__xeCJK_if_last_punct_glue:TF { \use_i:nn } }
4039   { \use:n }
4040   { \__xeCJK_last_punct_penalty_false:nn }
4041 }
4042 \cs_new_protected:Npn \__xeCJK_last_punct_penalty_false:nn #1#2
4043 {
4044   \bool_set_false:N \l__xeCJK_last_penalty_bool
4045   \tex_penalty:D \l__xeCJK_last_penalty_int
4046   #2
4047 }
4048 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxi:TF
4049 {
4050   \tex_unpenalty:D
4051   \bool_if:NF \l__xeCJK_last_penalty_bool
4052   {
4053     \bool_set_true:N \l__xeCJK_last_penalty_bool
4054     \int_set_eq:NN \l__xeCJK_last_penalty_int \c__xeCJK_nobreak_penalty_int
4055   }
4056   \xeCJK_if_last_node:TF
4057   { \__xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
4058   { \use:n }
4059   { \xeCJK_no_break: \use_ii:nn }
4060 }
4061 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxii:TF
4062 {
4063   \dim_compare:nNnTF \l__xeCJK_last_kern_dim > \c_zero_dim
4064   { \__xeCJK_if_last_punct_auxiii:TF }
4065   { \__xeCJK_make_node:N \l__xeCJK_last_kern_dim \use_ii:nn }
4066 }
4067 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiii:TF
4068 {
4069   \int_case:nnTF { \tex_XeTeXcharclass:D \l__xeCJK_last_kern_dim }
4070   {
4071     { \xeCJK_class_num:n { FullRight } }
4072     { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_right_tl }
4073     { \xeCJK_class_num:n { FullLeft } }
4074     { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_left_tl }
4075   }
4076   { \__xeCJK_if_last_punct_auxiv:TF }
4077   { \use_ii:nn }
4078 }
4079 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiv:TF
4080 {
4081   \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
4082   \xeCJK_if_last_node:TF
4083   {
4084     \tl_gset:Ne \g__xeCJK_last_punct_tl
4085     { \tex_Uchar:D \l__xeCJK_tmp_dim }
4086     \dim_set_eq:NN \l__xeCJK_last_bound_dim \l__xeCJK_last_kern_dim
4087     \use_i:nn
4088   }
4089   { \__xeCJK_make_node:N \l__xeCJK_tmp_dim \use_ii:nn }
4090 }
4091 \tl_new:N \l__xeCJK_aligni_tl
4092 \tl_new:N \l__xeCJK_alignii_tl
4093 \int_new:N \l__xeCJK_last_penalty_int
4094 \dim_new:N \l__xeCJK_last_bound_dim
4095 \bool_new:N \l__xeCJK_last_penalty_bool
\xeCJK_if_last_node:TF 4096 \cs_new_protected:Npn \xeCJK_if_last_node:TF #1#2
4097 {
4098   \__xeCJK_if_last_kern:TF

```

```

4099     {
4100         \dim_set_eq:NN \l__xeCJK_last_kern_dim \tex_lastkern:D
4101         \tex_unkern:D
4102         \__xeCJK_if_last_kern:TF
4103         {
4104             \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_last_kern_dim }
4105             { \tex_unkern:D #1 }
4106             { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
4107         }
4108         { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
4109     }
4110     {#2}
4111 }
4112 \dim_new:N \l__xeCJK_last_kern_dim

```

```

\__xeCJK_CJK_and_FullRight_glue:N 4113 \cs_new_protected:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
\__xeCJK_Default_and_FullRight_glue:N 4114 {
4115     \__xeCJK_punct_if_long:NTF #1
4116     {
4117         \__xeCJK_punct_if_nobreak_long:NTF #1
4118         { \xeCJK_no_break: }
4119         { \xeCJK_allow_break: }
4120     }
4121     { \xeCJK_no_break: }
4122     \__xeCJK_punct_if_middle:NT #1
4123     {
4124         \CJKglue
4125         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
4126         \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
4127     }
4128 }
4129 \cs_new_protected:Npn \__xeCJK_Default_and_FullRight_glue:N #1
4130 {
4131     \__xeCJK_punct_if_long:NTF #1
4132     {
4133         \__xeCJK_punct_if_nobreak_long:NTF #1
4134         { \xeCJK_no_break: }
4135         { \xeCJK_allow_break: }
4136     }
4137     { \xeCJK_no_break: }
4138     \__xeCJK_punct_if_middle:NT #1
4139     {
4140         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
4141         \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
4142     }
4143 }

```

```

\xeCJK_FullLeft_and_FullLeft:N 4144 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullLeft:N #1
4145 {
4146     \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
4147     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4148     \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4149     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4150     \xeCJK_fallback_punct_symbol:NN
4151     \CJKpunctsymbol #1
4152 }

```

```

\xeCJK_FullLeft_and_FullRight:N 4153 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullRight:N #1
4154 {
4155     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4156     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4157     \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4158     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4159     \xeCJK_FullRight_symbol:N #1
4160 }

```

```

\xeCJK_FullRight_and_FullLeft:N 4161 \cs_new_protected:Npn \xeCJK_FullRight_and_FullLeft:N #1
4162 {

```

```

4163 \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
4164 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4165 \xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4166 \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4167 \xeCJK_fallback_punct_symbol:NN
4168 \CJKpunctsymbol #1
4169 }

\xeCJK_FullRight_and_FullRight:N 4170 \cs_new_protected:Npn \xeCJK_FullRight_and_FullRight:N #1
4171 {
4172 \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
4173 \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
4174 \__xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
4175 \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
4176 \xeCJK_FullRight_symbol:N #1
4177 }

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```

4178 \keys_define:nn { xeCJK / options }
4179 {
4180   CheckFullRight .choice: ,
4181   CheckFullRight / true .code:n =
4182   {
4183     \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
4184     {
4185       \cs_set_eq:NN \__xeCJK_save_FullRight_check:
4186         \xeCJK_FullRight_and_Boundary:
4187       \cs_set_eq:NN \__xeCJK_save_FullRight_symbol:N
4188         \xeCJK_FullRight_symbol:N
4189       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
4190         \xeCJK_check_FullRight:
4191       \cs_set_eq:NN \xeCJK_FullRight_symbol:N
4192         \xeCJK_check_FullRight_symbol:Nw
4193     }
4194   } ,
4195   CheckFullRight / false .code:n =
4196   {
4197     \cs_if_eq:NNNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
4198     {
4199       \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
4200         \__xeCJK_save_FullRight_check:
4201       \cs_set_eq:NN \xeCJK_FullRight_symbol:N
4202         \__xeCJK_save_FullRight_symbol:N
4203     }
4204   } ,
4205   CheckFullRight .default:n = { true }
4206 }

```

```

experiment/halfright-prebreakpenalty 4207 \keys_define:nn { xeCJK / boundary-register }
experiment/punct-measure-fix 4208 {
experiment/boundary-register 4209   command .code:n =
4210     { \__xeCJK_boundary_user_register_command:n {#1} } ,
4211   command .value_required:n = true ,
4212   strategy .code:n =
4213     { \__xeCJK_boundary_user_register_strategy:n {#1} } ,
4214   strategy .value_required:n = true ,
4215   mode .code:n =
4216     { \__xeCJK_boundary_user_register_mode:n {#1} } ,
4217   mode .value_required:n = true
4218 }
4219 \keys_define:nn { xeCJK / options / experiment }
4220 {
4221   halfright-prebreakpenalty .bool_set:N = \l__xeCJK_halfright_nobreak_bool ,
4222   halfright-prebreakpenalty .default:n = true ,

```

```

4223     halfright-prebreakpenalty .initial:n = false ,
4224     punct-measure-fix .bool_gset:N = \g__xeCJK_punct_measure_fix_bool ,
4225     punct-measure-fix .default:n = true ,
4226     punct-measure-fix .initial:n = false ,
4227     boundary-register .code:n =
4228     { \__xeCJK_boundary_user_register:n {#1} } ,
4229     boundary-register .value_required:n = true
4230 }

\__xeCJK_FullRight_symbol:N 4231 \cs_new_protected:Npn \xeCJK_FullRight_symbol:N
4232 {
4233     \xeCJK_fallback_punct_symbol:NN
4234     \CJKpunctsymbol
4235 }

\__xeCJK_check_FullRight: 4236 \cs_new_protected:Npn \xeCJK_check_FullRight:
4237 {
4238     \xeCJK_get_punct_bounds:No \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4239     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4240     \group_align_safe_begin:
4241     \token_case_meaning:NoTF \l_peek_token
4242     { \l__xeCJK_no_break_cs_case_tl }
4243     {
4244         \group_align_safe_end:
4245         \xeCJK_no_break:
4246         \group_insert_after:N \xeCJK_no_break:
4247     }
4248     { \group_align_safe_end: }
4249     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
4250     \xeCJK_class_group_end:
4251     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
4252     \__xeCJK_punct_boundary_guard:
4253 }
4254 \prg_generate_conditional_variant:Nnn \token_case_meaning:Nn { No } { TF , F }

\__xeCJK_check_FullRight_symbol:Nw 4255 \cs_new_protected:Npn \xeCJK_check_FullRight_symbol:Nw #1
4256 { \peek_remove_spaces:n { \__xeCJK_save_FullRight_symbol:N #1 } }

\__xeCJK_cs_case_keys_define:nNnn 4257 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNnn #1#2#3#4#5
4258 {
4259     \tl_new:N #2
4260     \seq_new:N #3
4261     \keys_define:nn { xeCJK / options }
4262     {
4263         #1 .code:n =
4264         {
4265             \seq_set_split:Nnn #3 { } {##1}
4266             \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
4267         } ,
4268         #1+ .code:n =
4269         {
4270             \tl_map_inline:nn {##1}
4271             { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
4272             \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
4273         } ,
4274         #1- .code:n =
4275         {
4276             \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {####1} }
4277             \__xeCJK_update_cs_case_tl:Nnn #2#3 {#4} {#5}
4278         }
4279     }
4280 }
4281 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:Nnn #1#2#3#4
4282 {
4283     \tl_clear:N #1
4284     \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
4285     #4
4286 }

```

NoBreakCS 设置不能在全角右标点之后断行的控制序列。

```
4287 \xeCJK_cs_case_keys_define:nNnn { NoBreakCS }
4288 \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。

```
4289 \NewDocumentCommand \xeCJKnobreak { }
4290 {
4291   \bool_set_true:N \l__xeCJK_tmp_bool
4292   \int_while_do:nNnn \tex_lastnodetype:D = \c__xeCJK_glue_node
4293   {
4294     \bool_if:NTF \l__xeCJK_tmp_bool
4295     {
4296       \bool_set_false:N \l__xeCJK_tmp_bool
4297       \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
4298     }
4299     { \skip_add:Nn \l__xeCJK_last_skip \tex_lastskip:D }
4300     \tex_unskip:D
4301   }
4302   \xeCJK_if_last_node:TF
4303   {
4304     \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
4305     \xeCJK_if_last_node:TF
4306     {
4307       \__xeCJK_if_last_glue:T
4308       {
4309         \exp_args:NNNo \tex_unskip:D \xeCJK_no_break:
4310         \skip_horizontal:n { \skip_use:N \tex_lastskip:D }
4311       }
4312       \__xeCJK_make_node:N \l__xeCJK_last_kern_dim
4313     }
4314     { }
4315     \__xeCJK_make_node:N \l__xeCJK_tmp_dim
4316   }
4317   { }
4318   \xeCJK_no_break:
4319   \bool_if:NF \l__xeCJK_tmp_bool
4320   { \skip_horizontal:N \l__xeCJK_last_skip }
4321 }
```

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```
4322 \keys_define:nn { xeCJK / options }
4323 {
4324   CheckSingle .choice: ,
4325   CheckSingle / true .code:n =
4326   {
4327     \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4328     {
4329       \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
4330       \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4331     }
4332   } ,
4333   CheckSingle / false .code:n =
4334   {
4335     \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
4336     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
4337   } ,
4338   CheckSingle .default:n = { true } ,
4339   CJKchecksingle .meta:n = { CheckSingle = true }
4340 }
```

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

```
4341 \keys_define:nn { xeCJK / options }
4342 {
4343   WidowPenalty .int_set:N = \l__xeCJK_widow_penalty_int ,
4344   WidowPenalty .default:n = { 10 000 }
4345 }
```

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

```
4346 \cs_new_protected:Npn \xeCJK_widow_penalty:
4347 { \tex_penalty:D \l__xeCJK_widow_penalty_int }
```

```
\xeCJK_check_single:Nw 4348 \cs_new_protected:Npn \xeCJK_check_single:Nw #1
\_xeCJK_check_single_end:N 4349 {
4350   \group_align_safe_begin:
4351   \peek_catcode:NTF \c_catcode_letter_token
4352   { \xeCJK_check_single:NNw #1 }
4353   {
4354     \token_if_other:NTF \l_peek_token
4355     { \xeCJK_check_single:NNw }
4356     { \_xeCJK_check_single_end:N }
4357     #1
4358   }
4359 }
4360 \cs_new_protected:Npn \_xeCJK_check_single_end:N
4361 {
4362   \group_align_safe_end:
4363   \_xeCJK_check_single_save:N
4364 }
```

`\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:nNw`

```
4365 \cs_new_protected:Npn \xeCJK_check_single:NNw #1#2
4366 {
4367   \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
4368   {
4369     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
4370     {
4371       \bool_if:NTF \l__xeCJK_reserve_space_bool
4372       { \_xeCJK_check_single_end:N #1 #2 ~ }
4373       { \_xeCJK_check_single_space:NN #1#2 }
4374     }
4375     { \_xeCJK_check_single_end:N #1 #2 }
4376   }
4377   {
4378     \token_if_other:NTF \l_peek_token
4379     {
4380       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
4381       { \_xeCJK_check_single_space:NN }
4382       { \_xeCJK_check_single_end:N }
4383     }
4384     {
4385       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
4386       { \_xeCJK_check_single_aux:nNw { ~ } }
4387       { \_xeCJK_check_single_aux:nNw { } }
4388     }
4389     #1 #2
4390   }
4391 }
4392 \cs_new_protected:Npn \_xeCJK_check_single_aux:nNw #1#2#3
4393 {
4394   \token_if_cs:NTF \l_peek_token
4395   { \xeCJK_check_single_cs:NNn }
4396   { \xeCJK_check_single_end:NNnw }
4397   #2 #3 {#1}
4398 }
```

```

\XeCJK_check_single_end:NNnw 4399 \cs_new_protected:Npn \__XeCJK_check_single_end_aux:NNn #1#2#3
\__XeCJK_check_single_end_aux:NNn 4400 { \__XeCJK_check_single_end:N #1 #2 #3 }
\__XeCJK_check_single_end_equation:NNnw 4401 \cs_new_eq:NN \XeCJK_check_single_end:NNnw \__XeCJK_check_single_end_aux:NNn
4402 \cs_new_protected:Npn \__XeCJK_check_single_end_equation:NNnw
4403 {
4404   \token_if_math_toggle:NTF \l_peek_token
4405   { \XeCJK_check_single_equation:NNnNw }
4406   { \__XeCJK_check_single_end_aux:NNn }
4407 }

PlainEquation 4408 \keys_define:nn { XeCJK / options }
4409 {
4410   PlainEquation .choice: ,
4411   PlainEquation / true .code:n =
4412   {
4413     \cs_set_eq:NN \XeCJK_check_single_end:NNnw
4414     \__XeCJK_check_single_end_equation:NNnw
4415   } ,
4416   PlainEquation / false .code:n =
4417   {
4418     \cs_set_eq:NN \XeCJK_check_single_end:NNnw
4419     \__XeCJK_check_single_end_aux:NNn
4420   } ,
4421   PlainEquation .default:n = { true } ,
4422 }

\__XeCJK_check_single_space:NN 4423 \cs_new_protected:Npn \__XeCJK_check_single_space:NN #1#2
4424 {
4425   \XeCJK_if_CJK_class:NTF #2
4426   {
4427     \XeCJK_if_CJK_class:NTF \l_peek_token
4428     { \__XeCJK_check_single_end:N #1 #2 }
4429     { \__XeCJK_check_single_end:N #1 #2 ~ }
4430   }
4431   { \__XeCJK_check_single_end:N #1 #2 ~ }
4432 }

\XeCJK_check_single_equation:NNnNw 4433 \cs_new_protected:Npn \XeCJK_check_single_equation:NNnNw #1#2#3#4
4434 {
4435   \peek_catcode:NTF \c_math_toggle_token
4436   {
4437     \XeCJK_widow_penalty: \__XeCJK_check_single_end:N #1
4438     \XeCJK_make_node:n { CJK-widow } #2 #4
4439   }
4440   { \__XeCJK_check_single_end:N #1 #2#3#4 }
4441 }

```

\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。

```

4442 \cs_new_protected:Npn \XeCJK_check_single_cs:NNn #1#2#3
4443 {
4444   \token_case_meaning:Nof \l_peek_token
4445   { \l__XeCJK_check_single_cs_case_tl }
4446   { \use_iii:nnn }
4447   { \XeCJK_check_single_env:nnNn }
4448   {
4449     \XeCJK_widow_penalty:
4450     \__XeCJK_check_single_end:N #1

```

```

4451     \xeCJK_make_node:n { CJK-widow } #2#3
4452   }
4453   { \__xeCJK_check_single_end:N #1 #2#3 }
4454 }
4455 \tl_new:N \l__xeCJK_check_single_cs_case_tl

\xecjk_check_single_env:nnNn 4456 \cs_new_protected:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
4457 {
4458   \str_case_e:noTF {#4}
4459   { \l__xeCJK_inline_env_case_tl }
4460   {#2}
4461   {#1}
4462   #3 {#4}
4463 }
4464 \prg_generate_conditional_variant:Nnn \str_case_e:nn { no } { TF }

NewLineCS 4465 \xeCJK_cs_case_keys_define:nnNnn { NewLineCS }
4466   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
4467   { \use_ii:nnn }
4468   {
4469     \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
4470     \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
4471   }

EnvCS 4472 \xeCJK_cs_case_keys_define:nnNnn { EnvCS }
4473   \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
4474   { \use:n }
4475   {
4476     \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
4477     \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
4478   }

InlineEnv 4479 \keys_define:nn { xeCJK / options }
4480   {
4481     InlineEnv .code:n =
4482     {
4483       \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
4484       \__xeCJK_update_inline_env_case_tl:
4485     } ,
4486     InlineEnv+ .code:n =
4487     {
4488       \clist_map_inline:nn {#1}
4489       {
4490         \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
4491         { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
4492       }
4493       \__xeCJK_update_inline_env_case_tl:
4494     } ,
4495     InlineEnv- .code:n =
4496     {
4497       \clist_map_inline:nn {#1}
4498       { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
4499       \__xeCJK_update_inline_env_case_tl:
4500     }
4501   }
4502 \seq_new:N \l__xeCJK_inline_env_seq

\__xeCJK_update_inline_env_case_tl: 4503 \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
4504 {
4505   \tl_clear:N \l__xeCJK_inline_env_case_tl
4506   \seq_map_inline:Nn \l__xeCJK_inline_env_seq
4507   { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { } } }
4508 }
4509 \tl_new:N \l__xeCJK_inline_env_case_tl

```

5.9 增加 CJK 子分区

```

\g__xeCJK_CJK_sub_class_seq 4510 \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。

4511 \NewDocumentCommand \XeCJKDeclareSubCJKBlock
4512 { s > { \TrimSpaces } m m }
4513 {
4514   \XeCJK_declare_sub_char_class:nen { CJK } {#2} {#3}
4515   \IfBooleanT {#1} { \XeCJKResetPunctClass }
4516 }
4517 \@onlypreamble \XeCJKDeclareSubCJKBlock

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

4518 \bool_new:N \l__xeCJK_sub_cancel_bool
4519 \NewDocumentCommand \XeCJKCancelSubCJKBlock { s m }
4520 {
4521   \bool_if:NF \l__xeCJK_sub_cancel_bool
4522   {
4523     \bool_set_true:N \l__xeCJK_sub_cancel_bool
4524     \__xeCJK_sub_restore_or_cancel:e {#2}
4525     \IfBooleanT {#1} { \XeCJKResetPunctClass }
4526   }
4527 }
4528 \NewDocumentCommand \XeCJKRestoreSubCJKBlock { s m }
4529 {
4530   \bool_if:NT \l__xeCJK_sub_cancel_bool
4531   {
4532     \bool_set_false:N \l__xeCJK_sub_cancel_bool
4533     \__xeCJK_sub_restore_or_cancel:e {#2}
4534     \IfBooleanT {#1} { \XeCJKResetPunctClass }
4535   }
4536 }

\__xeCJK_sub_restore_or_cancel:n 4537 \cs_new_protected:Npn \__xeCJK_sub_restore_or_cancel:n #1
4538 {
4539   \clist_map_inline:nn {#1}
4540   {
4541     \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
4542     {
4543       \XeCJK_declare_char_class:nn
4544       { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
4545       { \use:c { g__xeCJK_CJK/##1_range_clist } }
4546     }
4547     { \__xeCJK_error:ne { SubBlock-undefined } {##1} }
4548   }
4549 }
4550 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { e }
4551 \__xeCJK_msg_new:nn { SubBlock-undefined }
4552 {
4553   The~CJK~sub~block~`#1'~is~undefined.\\
4554   Try~to~use~\token_to_str:N \XeCJKDeclareSubCJKBlock \
4555   to~declare~it.
4556 }

\XeCJK_declare_sub_char_class:nnn 4557 \cs_new_protected:Npn \XeCJK_declare_sub_char_class:nnn #1#2#3
4558 {
4559   \int_if_exist:cF { \__xeCJK_class_csname:n { #1/#2 } }
4560   {
4561     \XeCJK_new_class:n { #1/#2 }
4562     \__xeCJK_set_sub_class_toks:nn {#1} {#2}
4563     \XeCJK_new_sub_key:n {#2}
4564   }
4565   \XeCJK_declare_char_class:nn { #1/#2 } {#3}
4566 }
4567 \cs_generate_variant:Nn \XeCJK_declare_sub_char_class:nnn { ne }

```

```

\__xeCJK_set_sub_class_toks:nn 4568 \cs_new_protected:Npn \__xeCJK_set_sub_class_toks:nn #1#2
4569 {
4570   \seq_map_inline:Nn \g__xeCJK_base_class_seq
4571   {
4572     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
4573     \xeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
4574     \str_if_eq:nnTF {##1} { CJK }
4575     {
4576       \xeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
4577       { \__xeCJK_switch_font:nn {#1} {#2} }
4578     }
4579     {
4580       \xeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
4581       { \xeCJK_fallback_symbol:NN }
4582       {
4583         \__xeCJK_switch_font:nn {#1} {#2}
4584         \xeCJK_fallback_symbol:NN
4585       }
4586     }
4587   }
4588   \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
4589   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
4590   {
4591     \xeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
4592     \xeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
4593     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
4594     { \__xeCJK_switch_font:nn {#2} {##1} }
4595     \xeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
4596     { \__xeCJK_switch_font:nn {##1} {#2} }
4597   }
4598   \seq_gput_right:Nn \g__xeCJK_CJK_sub_class_seq {#2}
4599   \__xeCJK_save_CJK_class:n { #1/#2 }
4600   \clist_map_inline:nn
4601   {
4602     CJK , FullLeft , FullRight , HangulJamo ,
4603     HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
4604   }
4605   {
4606     \xeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
4607     { \__xeCJK_switch_font:nn {#2} {#1} }
4608   }
4609 }

```

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。

```

4610 \cs_if_exist:NF \tex_XeTeXglyphbounds:D
4611 {
4612   \__xeCJK_msg_new:nn { XeTeX-too-old }
4613   {
4614     \token_to_str:N \tex_XeTeXglyphbounds:D \ is~not~defined.\\
4615     CJK~punctuation~kerning~will~not~be~available.\\
4616     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
4617   }
4618   \__xeCJK_error:n { XeTeX-too-old }
4619   \AtEndOfPackage
4620   {
4621     \keys_define:nn { xeCJK / options }
4622     {
4623       PunctStyle .code:n =
4624       { \__xeCJK_error:ne { punct-style-unknown } {#1} }
4625     }
4626     \seq_gclear:N \g__xeCJK_punct_style_seq
4627     \__xeCJK_set_punct_style:n { plain }

```

```

4628     }
4629 }

```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```

4630 \NewDocumentCommand \xeCJKsetwidth { s m m }
4631 {
4632   \IfBooleanTF {#1}
4633   {
4634     \tl_map_inline:en {#2}
4635     {
4636       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
4637       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
4638       \tl_gset:cn { g__xeCJK_punct_bound_width/##1/tl } {#3}
4639     }
4640   }
4641   {
4642     \tl_map_inline:en {#2}
4643     {
4644       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
4645       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
4646       \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#3}
4647     }
4648   }
4649 }
4650 \@onlypreamble \xeCJKsetwidth
4651 \cs_generate_variant:Nn \tl_map_inline:nn { e }

```

\xeCJKsetkern 手动设置相邻标点的距离。

```

4652 \NewDocumentCommand \xeCJKsetkern { m m m }
4653 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
4654 \@onlypreamble \xeCJKsetkern

\c__xeCJK_left_tl 4655 \tl_const:Nn \c__xeCJK_left_tl { left }
\c__xeCJK_right_tl 4656 \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

4657 \keys_define:nn { xeCJK / options }
4658 {
4659   AllowBreakBetweenPuncts .choice: ,
4660   AllowBreakBetweenPuncts / true .code:n =
4661   {
4662     \bool_set_true:N \l__xeCJK_punct_breakable_bool
4663     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
4664     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
4665     \__xeCJK_punct_bound_breakable_kern:N
4666   } ,
4667   AllowBreakBetweenPuncts / false .code:n =
4668   {
4669     \bool_set_false:N \l__xeCJK_punct_breakable_bool
4670     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_kern:NN
4671     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
4672     \__xeCJK_nobreak_hskip:N
4673   } ,
4674   AllowBreakBetweenPuncts .default:n = { true } ,
4675   KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
4676   KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
4677   KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
4678   LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
4679   LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
4680   LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
4681   NoBreakLongPunct .code:n = { \__xeCJK_set_special_punct:nn { nobreak_long } {#1} } ,
4682   NoBreakLongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { nobreak_long } {#1} } ,
4683   NoBreakLongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { nobreak_long } {#1} } ,
4684   MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
4685   MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,

```

```

4686 MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
4687 PoZheHaoLigature .choice: ,
4688 PoZheHaoLigature / true .code:n =
4689 {
4690   \bool_set_true:N \l__xeCJK_pozhehao_ligature_bool
4691   \xeCJK_declare_char_class:nn { PoZheHao } { "2014" , "2015" }
4692 } ,
4693 PoZheHaoLigature / false .code:n =
4694 {
4695   \bool_set_false:N \l__xeCJK_pozhehao_ligature_bool
4696   \xeCJK_declare_char_class:nn { FullRight } { "2014" }
4697   \xeCJK_declare_char_class:nn { Default } { "2015" }
4698 } ,
4699 PoZheHaoLigature .default:n = { true } ,
4700 LatinPunct .choice: ,
4701 LatinPunct / true .code:n =
4702 {
4703   \bool_set_true:N \l__xeCJK_latin_punct_bool
4704   \xeCJK_declare_char_class:nn { HalfLeft }
4705   { "2018" , "201C" }
4706   \xeCJK_declare_char_class:nn { HalfRight }
4707   { "00B7" , "2019" , "201D" , "2025" , "2026" , "2027" }
4708 } ,
4709 LatinPunct / false .code:n =
4710 {
4711   \bool_set_false:N \l__xeCJK_latin_punct_bool
4712   \xeCJK_declare_char_class:nn { FullLeft }
4713   { "2018" , "201C" }
4714   \xeCJK_declare_char_class:nn { FullRight }
4715   { "00B7" , "2019" , "201D" , "2025" , "2026" , "2027" }
4716 } ,
4717 LatinPunct .default:n = { true } ,
4718 CJLineBreak .choices:nn = { normal , strict }
4719 {
4720   \str_if_eq:nnTF {#1} { strict }
4721   {
4722     \bool_set_true:N \l__xeCJK_CJ_strict_bool
4723     \xeCJK_declare_char_class:nN { CJStarter } \c__xeCJK_CJ_chars_clist
4724   }
4725   {
4726     \bool_set_false:N \l__xeCJK_CJ_strict_bool
4727     \xeCJK_declare_char_class:nN { CJK } \c__xeCJK_CJ_chars_clist
4728   }
4729 } ,
4730 CJLineBreak .initial:n = { normal } ,
4731 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
4732 PunctBoundWidth.tl_gset:N = \g__xeCJK_punct_bound_width_tl ,
4733 PunctWidth .value_required:n = true ,
4734 PunctBoundWidth .value_required:n = true ,
4735 RubberPunctSkip .choice: ,
4736 RubberPunctSkip .default:n = { true } ,
4737 RubberPunctSkip / true .code:n =
4738 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN } ,
4739 RubberPunctSkip / plus .code:n =
4740 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_plus:nNN } ,
4741 RubberPunctSkip / minus .code:n =
4742 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_minus:nNN } ,
4743 RubberPunctSkip / false .code:n =
4744 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_dim:nNN }
4745 }
4746 \bool_new:N \l__xeCJK_punct_breakable_bool

```

相关选项定义的辅助函数。

```

4747 \clist_new:N \g__xeCJK_special_punct_clist
4748 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , nobreak_long , middle }
4749 \cs_new:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
4750 \cs_new:Npn \__xeCJK_special_punct_tl:n #1#2 { g__xeCJK_special_punct_#1_#2_tl }

```

```

4751 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
4752 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
4753 \cs_new_protected:Npn \__xeCJK_set_special_punct:nn #1#2
4754 {
4755   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
4756   { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
4757   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
4758   \tl_map_inline:en {#2}
4759   {
4760     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4761     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4762   }
4763 }
4764 \cs_new_protected:Npn \__xeCJK_add_special_punct:nn #1#2
4765 {
4766   \tl_map_inline:en {#2}
4767   {
4768     \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
4769     {
4770       \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4771       \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4772     }
4773   }
4774 }
4775 \cs_new_protected:Npn \__xeCJK_sub_special_punct:nn #1#2
4776 {
4777   \tl_map_inline:en {#2}
4778   {
4779     \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
4780     \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
4781   }
4782 }

```

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。

```

4783 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
4784 {
4785   \if_int_compare:w \xeCJK_token_value_class:N #1 =
4786     \xeCJK_class_num:n { FullRight }
4787     \prg_return_true:
4788   \else:
4789     \if_int_compare:w \xeCJK_token_value_class:N #1 =
4790       \xeCJK_class_num:n { PoZheHao }
4791       \prg_return_true:
4792     \else:
4793       \prg_return_false:
4794     \fi:
4795   \fi:
4796 }
4797 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
4798 {
4799   \exp_args:Nc
4800   \prg_new_conditional:Npnn { __xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
4801   {
4802     \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
4803     \prg_return_true: \else: \prg_return_false: \fi:
4804   }
4805 }

```

一些用于记录的辅助函数。

```

4806 \cs_new:Npn \__xeCJK_punct_csname:n #1
4807 { c__xeCJK_\l_xeCJK_current_punct_font_tl/\l_xeCJK_punct_style_tl/#1/tl }
4808 \cs_new:Npn \__xeCJK_use_punct_dim:nN #1#2
4809 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2 } } }
4810 \cs_new:Npn \__xeCJK_use_punct_dim:nNN #1#2#3
4811 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2/#3 } } }

```

```

4812 \cs_new:Npn \__xeCJK_use_punct_skip:nNN #1#2#3
4813 { \use:c { \__xeCJK_punct_csname:n { skip/#1/#2/#3 } } }
4814 \cs_new:Npn \__xeCJK_use_punct_skip_plus:nNN #1#2#3
4815 { \use:c { \__xeCJK_punct_csname:n { skip/plus/#1/#2/#3 } } }
4816 \cs_new:Npn \__xeCJK_use_punct_skip_minus:nNN #1#2#3
4817 { \use:c { \__xeCJK_punct_csname:n { skip/minus/#1/#2/#3 } } }
4818 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNn #1#2
4819 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2 } }
4820 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNNn #1#2#3
4821 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2/#3 } }
4822 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNn #1#2#3#4
4823 {
4824   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { #1/#2/#3 } {#4}
4825   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { plus/#1/#2/#3 } {#4}
4826   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { minus/#1/#2/#3 } {#4}
4827 }
4828 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNnnn #1#2#3#4#5#6
4829 {
4830   \exp_last_unbraced:Ne
4831   \__xeCJK_save_punct_skip_aux:nnnnn
4832   {
4833     {#1}
4834     { #1/#2/#3 }
4835     { \dim_eval:n {#4} }
4836     { \dim_max:nn { \c_zero_dim } {#5} }
4837     { \dim_max:nn { \c_zero_dim } {#6} }
4838   }
4839 }
4840 \cs_new_protected:Npn \__xeCJK_save_punct_skip_aux:nnnnn #1#2#3#4#5
4841 {
4842   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4843   {#2} { #3 ~ plus ~ #4 ~ minus ~ #5 ~ }
4844   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4845   { plus/#2 } { #3 ~ plus ~ #4 ~ }
4846   \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
4847   { minus/#2 } { #3 ~ minus ~ #5 ~ }
4848 }
4849 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:nnnn #1#2#3#4
4850 {
4851   \__xeCJK_save_punct_width_aux:cen
4852   { \__xeCJK_punct_csname:n { #1/#3 } }
4853   { \use:c { #1_eval:n } {#4} }
4854   {#2}
4855 }
4856 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:Nnn #1#2#3
4857 {
4858   \tl_const:Nn #1 {#2}
4859   \str_if_eq:nnT {#3} { glue }
4860   { \prop_gput:Nnn \g__xeCJK_punct_skip_prop {#2} { } }
4861 }
4862 \prop_new:N \g__xeCJK_punct_skip_prop
4863 \prop_gput:Non \g__xeCJK_punct_skip_prop { \skip_use:N \c_zero_skip } { }
4864 \cs_generate_variant:Nn \__xeCJK_save_punct_width_aux:Nnn { ce }
4865 \cs_new_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN

```

定义标点处理模板。

```

4866 \DeclareObjectType { xeCJK / punctuation } { 0 }
4867 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { 0 }
4868 {
4869   enabled-global-setting      : boolean = true ,
4870   fixed-punct-width          : length  = \c_max_dim ,
4871   fixed-punct-ratio          : real    = \c_one_fp ,
4872   mixed-punct-width          : length  = \KeyValue { fixed-punct-width } ,
4873   mixed-punct-ratio          : real    = \KeyValue { fixed-punct-ratio } ,
4874   middle-punct-width         : length  = \KeyValue { fixed-punct-width } ,
4875   middle-punct-ratio         : real    = \KeyValue { fixed-punct-ratio } ,

```

```

4876     fixed-margin-width           : length = \c_max_dim ,
4877     fixed-margin-ratio           : real   = \c_one_fp ,
4878     mixed-margin-width          : length = \KeyValue { fixed-margin-width } ,
4879     mixed-margin-ratio          : real   = \KeyValue { fixed-margin-ratio } ,
4880     middle-margin-width         : length = \KeyValue { fixed-margin-width } ,
4881     middle-margin-ratio         : real   = \KeyValue { fixed-margin-ratio } ,
4882     bound-punct-width           : length = \c_max_dim ,
4883     bound-punct-ratio           : real   = \c_nan_fp ,
4884     bound-margin-width          : length = \c_max_dim ,
4885     bound-margin-ratio          : real   = \c_zero_fp ,
4886     enabled-hanging              : boolean = false ,
4887     add-min-bound-to-margin      : boolean = false ,
4888     optimize-margin              : boolean = false ,
4889     margin-minimum              : length = \c_zero_dim ,
4890     enabled-kerning              : boolean = true ,
4891     enabled-left-right-kerning   : boolean = true ,
4892     min-bound-to-kerning         : boolean = false ,
4893     kerning-total-width          : length = \c_max_dim ,
4894     kerning-total-ratio          : real   = 0.75 ,
4895     optimize-kerning             : boolean = false ,
4896     same-align-margin            : length = \c_max_dim ,
4897     same-align-ratio            : real   = \c_nan_fp ,
4898     different-align-margin       : length = \c_max_dim ,
4899     different-align-ratio        : real   = \c_nan_fp ,
4900     kerning-margin-width         : length = \c_max_dim ,
4901     kerning-margin-ratio         : real   = \c_one_fp ,
4902     kerning-margin-minimum       : length = \c_zero_dim
4903 }

4904 \DeclareTemplateCode { xeCJK / punctuation } { basic } { 0 }
4905 {
4906     enabled-global-setting        = \l__xeCJK_enabled_global_setting_bool ,
4907     fixed-punct-width            = \l__xeCJK_fixed_punct_width_dim ,
4908     fixed-punct-ratio            = \l__xeCJK_fixed_punct_ratio_fp ,
4909     mixed-punct-width            = \l__xeCJK_mixed_punct_width_dim ,
4910     mixed-punct-ratio            = \l__xeCJK_mixed_punct_ratio_fp ,
4911     middle-punct-width           = \l__xeCJK_middle_punct_width_dim ,
4912     middle-punct-ratio           = \l__xeCJK_middle_punct_ratio_fp ,
4913     fixed-margin-width           = \l__xeCJK_fixed_margin_width_dim ,
4914     fixed-margin-ratio           = \l__xeCJK_fixed_margin_ratio_fp ,
4915     mixed-margin-width           = \l__xeCJK_mixed_margin_width_dim ,
4916     mixed-margin-ratio           = \l__xeCJK_mixed_margin_ratio_fp ,
4917     middle-margin-width          = \l__xeCJK_middle_margin_width_dim ,
4918     middle-margin-ratio          = \l__xeCJK_middle_margin_ratio_fp ,
4919     bound-punct-width            = \l__xeCJK_bound_punct_width_dim ,
4920     bound-punct-ratio            = \l__xeCJK_bound_punct_ratio_fp ,
4921     bound-margin-width           = \l__xeCJK_bound_margin_width_dim ,
4922     bound-margin-ratio           = \l__xeCJK_bound_margin_ratio_fp ,
4923     enabled-hanging              = \l__xeCJK_enabled_hanging_bool ,
4924     add-min-bound-to-margin      = \l__xeCJK_add_min_bound_to_margin_bool ,
4925     optimize-margin              = \l__xeCJK_optimize_margin_bool ,
4926     margin-minimum               = \l__xeCJK_margin_minimum_dim ,
4927     enabled-kerning              = \l__xeCJK_enabled_kerning_bool ,
4928     enabled-left-right-kerning   = \l__xeCJK_enabled_left_right_kerning_bool ,
4929     min-bound-to-kerning         = \l__xeCJK_min_bound_to_kerning_bool ,
4930     kerning-total-width          = \l__xeCJK_kerning_total_width_dim ,
4931     kerning-total-ratio          = \l__xeCJK_kerning_total_ratio_fp ,
4932     optimize-kerning             = \l__xeCJK_optimize_kerning_bool ,
4933     same-align-margin            = \l__xeCJK_same_align_margin_dim ,
4934     same-align-ratio             = \l__xeCJK_same_align_ratio_fp ,
4935     different-align-margin       = \l__xeCJK_different_align_margin_dim ,
4936     different-align-ratio        = \l__xeCJK_different_align_ratio_fp ,
4937     kerning-margin-width         = \l__xeCJK_kerning_margin_width_dim ,
4938     kerning-margin-ratio         = \l__xeCJK_kerning_margin_ratio_fp ,
4939     kerning-margin-minimum       = \l__xeCJK_kerning_margin_minimum_dim
4940 }
4941 { \AssignTemplateKeys }

```

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

```

4942 \cs_new_protected:Npn \xeCJK_get_punct_bounds:NN #1#2
4943 {
4944   \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/glue/#1/#2 } }
4945   { \__xeCJK_get_punct_bounds_aux:NN #1 #2 }
4946 }
4947 \cs_new_protected:Npn \__xeCJK_get_punct_bounds_aux:NN
4948 {
4949   \tl_if_eq:NNTF \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
4950   { \__xeCJK_save_punct_margin_plain:NN }
4951   { \__xeCJK_save_punct_margin:NN }
4952 }
4953 \cs_new_protected:Npn \xeCJK_get_punct_bounds:No
4954 { \exp_last_unbraced:NNo \xeCJK_get_punct_bounds:NN }
4955 \cs_new_protected:Npn \__xeCJK_save_punct_margin_plain:NN #1#2
4956 {
4957   \__xeCJK_save_punct_dim:nNNn { glue } #1 #2 { \c_zero_dim }
4958   \__xeCJK_save_punct_dim:nNNn { offset } #1 #2 { \c_zero_dim }
4959   \__xeCJK_save_punct_dim:nNNn { margin } #1 #2 { \c_zero_dim }
4960   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_left_tl {#2} { \c_zero_dim }
4961   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_right_tl {#2} { \c_zero_dim }
4962   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
4963   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
4964   \__xeCJK_save_punct_skip:nNNn { glue } #1 #2 { \c_zero_skip }
4965 }
4966 \cs_new_protected:Npn \__xeCJK_save_punct_margin:NN #1#2
4967 {
4968   \group_begin:
4969   \xeCJK_select_punct_font:
4970   \xeCJK_fallback_punct_symbol:NN
4971   \xeCJK_calc_punct_dimen:N #2
4972   \group_end:
4973   \dim_set:Nn \l__xeCJK_bound_dim
4974   { \__xeCJK_use_punct_dim:nNN { bound } #1 #2 }
4975   \tl_if_eq:NNTF #1 \c__xeCJK_right_tl
4976   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_left_tl }
4977   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_right_tl }
4978   \dim_set:Nn \l__xeCJK_reverse_bound_dim
4979   { \__xeCJK_use_punct_dim:nNN { bound } \l__xeCJK_reverse_tl #2 }
4980   \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
4981   \xeCJK_punct_margin_process:NN #1 #2
4982   \xeCJK_punct_offset_process:NN #1 #2
4983   \__xeCJK_punct_if_long:NT #2
4984   { \__xeCJK_long_punct_kerning:N #2 }
4985 }
4986 \tl_new:N \l__xeCJK_reverse_tl
4987 \dim_new:N \l__xeCJK_bound_dim
4988 \dim_new:N \l__xeCJK_reverse_bound_dim

```

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

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

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

```

4989 \cs_new_protected:Npn \__xeCJK_long_punct_kerning:N #1
4990 {
4991   \dim_set:Nn \l__xeCJK_tmp_dim
4992   {
4993     \dim_max:n

```

```

4994         { \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim }
4995         { \c_zero_dim }
4996     }
4997     \l__xeCJK_save_punct_dim:nNNn { bound_width } #1 #1 { \l__xeCJK_tmp_dim }
4998     \dim_set:Nn \l__xeCJK_tmp_dim
4999     {
5000         \str_case:nnF {#1}
5001         {
5002             { ~~~~2014 }
5003             {
5004                 - \dim_max:nn
5005                 { \l__xeCJK_tmp_dim }
5006                 {
5007                     \dim_max:nn
5008                     {
5009                         \l__xeCJK_use_punct_dim:nN { dimen } #1 +
5010                         \l__xeCJK_use_punct_dim:nN { width } #1 -
5011                         \f@size pt - \f@size pt
5012                     }
5013                     {
5014                         \l__xeCJK_use_punct_dim:nN { width } #1 +
5015                         \l__xeCJK_use_punct_dim:nN { width } #1 -
5016                         \f@size pt - \f@size pt
5017                     }
5018                 }
5019             }
5020             { ~~~~2025 } { \c_zero_dim }
5021             { ~~~~2026 } { \c_zero_dim }
5022         }
5023         { -\l__xeCJK_tmp_dim }
5024     }
5025     \l__xeCJK_save_punct_dim:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
5026     \l__xeCJK_save_punct_skip:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
5027     \dim_add:Nn \l__xeCJK_tmp_dim
5028     { \dim_max:nn { \l__xeCJK_bound_dim } { \c_zero_dim } }
5029     \l__xeCJK_save_punct_dim:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
5030     \l__xeCJK_save_punct_skip:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
5031 }

```

\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
5032 \cs_new_protected:Npn \xeCJK_get_punct_kerning:NN #1#2
5033 {
5034     \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/kern/#1/#2 } }
5035     {
5036         \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
5037         { \__xeCJK_save_punct_kerning_plain:NN }
5038         { \__xeCJK_save_punct_kerning:NN }
5039         #1 #2
5040     }
5041 }
5042 \cs_new_protected:Npn \xeCJK_get_punct_kerning:oN
5043 { \exp_after:wN \xeCJK_get_punct_kerning:NN }
5044 \cs_new_protected:Npn \__xeCJK_save_punct_kerning_plain:NN #1#2
5045 {
5046     \l__xeCJK_save_punct_dim:nNNn { kern } #1 #2 { \c_zero_dim }
5047     \l__xeCJK_save_punct_dim:nNNn { bound_kern } #1 #2 { \c_zero_dim }
5048     \l__xeCJK_save_punct_dim:nNNn { bound_width } #1 #2 { \c_zero_dim }
5049     \l__xeCJK_save_punct_skip:nNNn { kern } #1 #2 { \c_zero_skip }
5050     \l__xeCJK_save_punct_skip:nNNn { bound_kern } #1 #2 { \c_zero_skip }
5051 }
5052 \cs_new_protected:Npn \__xeCJK_save_punct_kerning:NN #1#2
5053 {
5054     % 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。
5055     \UseInstance { xeCJK / punctuation } { \l__xeCJK_punct_style_tl }
5056     \__xeCJK_punct_if_right:NTF #1

```

```

5057     { \xeCJK_punct_kerning_process:NN #1 #2 }
5058     {
5059         \__xeCJK_punct_if_right:NTF #2
5060         { \__xeCJK_save_left_right_kerning:NN #1 #2 }
5061         { \xeCJK_punct_kerning_process:NN #1 #2 }
5062     }
5063 }
5064 \cs_new_protected:Npn \__xeCJK_save_left_right_kerning:NN #1#2
5065 {
5066     \bool_if:NTF \l__xeCJK_enabled_left_right_kerning_bool
5067     { \xeCJK_punct_kerning_process:NN #1 #2 }
5068     {
5069         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5070         {
5071             \tl_if_exist:cTF { g__xeCJK_punct/kern/#1/#2/tl }
5072             { \xeCJK_punct_kerning_process:NN #1 #2 }
5073             { \__xeCJK_save_punct_kerning_plain:NN #1 #2 }
5074         }
5075         { \__xeCJK_save_punct_kerning_plain:NN #1 #2 }
5076     }
5077 }

```

_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。

```

5078 \prg_new_conditional:Npnn \__xeCJK_punct_if_full_margin_dash:N #1 { p , T , F , TF }
5079 {
5080     \str_if_eq:nnTF {#1} { ~~~~2014 }
5081     {
5082         \if_int_compare:w \xeCJK_token_value_class:N #1 =
5083             \xeCJK_class_num:n { PoZheHao }
5084         \prg_return_false:
5085     }
5086     \else:
5087         \prg_return_true:
5088     \fi:
5089     { \prg_return_false: }
5090 }

```

\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。

```

5091 \cs_new_protected:Npn \xeCJK_punct_margin_process:NN #1#2
5092 {
5093     \dim_set:Nn \l__xeCJK_tmp_dim
5094     {
5095         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5096         {
5097             \cs_if_exist_use:cF { g__xeCJK_punct_width/#2/tl }
5098             {
5099                 \tl_if_empty:NTF \g__xeCJK_punct_width_tl
5100                 { \__xeCJK_calc_punct_width:N #2 }
5101                 { \g__xeCJK_punct_width_tl }
5102             }
5103         }
5104         { \__xeCJK_calc_punct_width:N #2 }
5105     }
5106     \dim_set:Nn \l__xeCJK_margin_dim
5107     {
5108         \dim_max:nn
5109         { \l__xeCJK_margin_minimum_dim }
5110         {
5111             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
5112             {
5113                 \__xeCJK_punct_if_middle:NTF #2
5114                 {

```

```

5115         ( \l__xeCJK_tmp_dim
5116         - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5117         )
5118         \__xeCJK_punct_if_full_margin_dash:Nf #2 { / 2 }
5119     }
5120     {
5121         \bool_if:NTF \l__xeCJK_optimize_margin_bool
5122         {
5123             \dim_max:nn
5124             {
5125                 \dim_min:nn
5126                 { \l__xeCJK_bound_dim }
5127                 { \l__xeCJK_reverse_bound_dim }
5128             }
5129         }
5130         { \use:n }
5131         {
5132             \l__xeCJK_tmp_dim
5133             - \l__xeCJK_reverse_bound_dim
5134             - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5135         }
5136     }
5137 }
5138 {
5139     \bool_if:NTF \l__xeCJK_optimize_margin_bool
5140     { \dim_min:nn { \l__xeCJK_bound_dim } }
5141     { \use:n }
5142     { \__xeCJK_calc_margin_width:N #2 }
5143 }
5144 }
5145 }
5146 \__xeCJK_save_punct_dim:nNn { margin } #1 #2 { \l__xeCJK_margin_dim }
5147 }
5148 \dim_new:N \l__xeCJK_margin_dim
\__xeCJK_calc_punct_width:N 5149 \cs_new:Npn \__xeCJK_calc_punct_width:N #1
5150 {
5151     \__xeCJK_punct_if_middle:Nf #1
5152     { \__xeCJK_punct_width_or_ratio:nN { middle } }
5153     {
5154         \__xeCJK_punct_if_mixed_width:Nf #1
5155         { \__xeCJK_punct_width_or_ratio:nN { mixed } }
5156         { \__xeCJK_punct_width_or_ratio:nN { fixed } }
5157     }
5158     #1
5159 }
\__xeCJK_calc_margin_width:N 5160 \cs_new:Npn \__xeCJK_calc_margin_width:N #1
5161 {
5162     \__xeCJK_punct_if_middle:Nf #1
5163     {
5164         \dim_compare:nNf \l__xeCJK_middle_margin_width_dim < \c_max_dim
5165         { \l__xeCJK_middle_margin_width_dim }
5166         {
5167             \__xeCJK_dim_ratio:Nn \l__xeCJK_middle_margin_ratio_fp
5168             { ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / 2 }
5169         }
5170     }
5171     {
5172         \__xeCJK_punct_if_mixed_width:Nf #1
5173         { \__xeCJK_margin_width_or_ratio:n { mixed } }
5174         { \__xeCJK_margin_width_or_ratio:n { fixed } }
5175     }
5176 }
\__xeCJK_dim_ratio:Nn 5177 \cs_new:Npn \__xeCJK_dim_ratio:Nn #1#2
5178 { \fp_to_dim:n { #1 \dim_to_fp:n {#2} } }
5179 \cs_generate_variant:Nn \__xeCJK_dim_ratio:Nn { c }

```

```

\xCJK_punct_offset_process:NN 5180 \cs_new_protected:Npn \xeCJK_punct_offset_process:NN #1#2
5181 {
5182   \dim_set:Nn \l__xeCJK_tmp_dim
5183   {
5184     \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5185     {
5186       \cs_if_exist_use:cF { g__xeCJK_punct_bound_width/#2/tl }
5187       {
5188         \tl_if_empty:NTF \g__xeCJK_punct_bound_width_tl
5189         { \__xeCJK_punct_width_or_ratio:nN { bound } #2 }
5190         { \g__xeCJK_punct_bound_width_tl }
5191       }
5192     }
5193     { \__xeCJK_punct_width_or_ratio:nN { bound } #2 }
5194   }
5195   \dim_set:Nn \l__xeCJK_tmp_dim
5196   {
5197     \bool_if:NTF \l__xeCJK_enabled_hanging_bool
5198     { \use:n }
5199     { \dim_max:nn { \l__xeCJK_margin_minimum_dim } }
5200     {
5201       \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
5202       {
5203         \__xeCJK_punct_if_middle:NTF #2
5204         {
5205           \l__xeCJK_tmp_dim
5206           - \l__xeCJK_margin_dim
5207           - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5208         }
5209         {
5210           \l__xeCJK_tmp_dim
5211           - \l__xeCJK_reverse_bound_dim
5212           - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
5213         }
5214       }
5215       {
5216         \bool_if:NTF \l__xeCJK_optimize_margin_bool
5217         { \dim_min:nn { \l__xeCJK_bound_dim } }
5218         { \use:n }
5219         { \__xeCJK_margin_width_or_ratio:n { bound } }
5220       }
5221     }
5222   }
5223   \__xeCJK_save_punct_dim:nNNn { offset } #1 #2
5224   { \l__xeCJK_tmp_dim }
5225   \__xeCJK_save_punct_dim:nNNn { rule } #1 #2
5226   { \l__xeCJK_tmp_dim - \l__xeCJK_bound_dim }
5227   \__xeCJK_save_punct_dim:nNNn { rule } \l__xeCJK_reverse_tl #2
5228   { \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim }
5229   \__xeCJK_save_punct_dim:nNNn { glue } #1 #2
5230   { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
5231   \__xeCJK_save_punct_skip:nNNnn { glue } #1 #2
5232   { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
5233   {
5234     \__xeCJK_punct_if_middle:NTF #2
5235     {
5236       ( \__xeCJK_use_punct_dim:nN { width } #2 -
5237         \__xeCJK_use_punct_dim:nN { dimen } #2 )
5238       \__xeCJK_punct_if_full_margin_dash:NF #2 { / 2 }
5239       - \l__xeCJK_margin_dim
5240     }
5241     { \l__xeCJK_bound_dim - \l__xeCJK_margin_dim }
5242   }
5243   {
5244     \__xeCJK_punct_if_middle:NTF #2
5245     { \l__xeCJK_margin_dim / 2 }
5246     { \l__xeCJK_margin_dim - \l__xeCJK_reverse_bound_dim }

```

```
5247     }
5248 }
```

```
\_xeCJK_punct_width_or_ratio:nN 5249 \cs_new:Npn \_xeCJK_punct_width_or_ratio:nN #1#2
5250 {
5251     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
5252     { \use:c { l__xeCJK_#1_punct_width_dim } }
5253     {
5254         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_punct_ratio_fp } }
5255         { \c_max_dim }
5256         {
5257             \_xeCJK_dim_ratio:cn
5258             { l__xeCJK_#1_punct_ratio_fp }
5259             { \_xeCJK_use_punct_dim:nN { width } #2 }
5260         }
5261     }
5262 }
```

```
\_xeCJK_margin_width_or_ratio:n 5263 \cs_new:Npn \_xeCJK_margin_width_or_ratio:n #1
5264 {
5265     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
5266     { \use:c { l__xeCJK_#1_margin_width_dim } }
5267     {
5268         \_xeCJK_dim_ratio:cn
5269         { l__xeCJK_#1_margin_ratio_fp }
5270         { \l__xeCJK_bound_dim }
5271     }
5272     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
5273     { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
5274 }
```

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

```
5275 \cs_new_protected:Npn \xeCJK_punct_kerning_process:NN #1#2
5276 {
5277     \dim_set:Nn \l__xeCJK_margin_dim
5278     { \_xeCJK_original_kerning_margin:NN #1 #2 }
5279     \dim_set:Nn \l__xeCJK_minimum_bound_dim
5280     { \_xeCJK_punct_min_bound:NN #1 #2 }
5281     \_xeCJK_punct_if_long:NTF #1
5282     { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
5283     {
5284         \_xeCJK_punct_if_long:NT #2
5285         { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
5286     }
5287     \dim_set:Nn \l__xeCJK_kerning_margin_dim
5288     {
5289         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
5290         {
5291             \cs_if_exist_use:cF { g__xeCJK_punct/kern/#1/#2/t1 }
5292             { \_xeCJK_punct_kerning_process_aux:NN #1 #2 }
5293         }
5294         { \_xeCJK_punct_kerning_process_aux:NN #1 #2 }
5295     }
5296     \_xeCJK_save_kerning:nnNN { kern } { bound } #1 #2
5297     \_xeCJK_save_punct_dim:nnNN { bound_width } #1 #2
5298     { \l__xeCJK_kerning_margin_dim - \l__xeCJK_tmp_dim }
5299     \_xeCJK_punct_if_right:NTF #1
5300     {
5301         \_xeCJK_punct_if_right:NTF #2
5302         { \_xeCJK_save_kerning:nnnNN { bound_kern } { offset } { bound } }
5303         { \_xeCJK_save_kerning:nnNN { bound_kern } { offset } }
5304     }
5305     {
5306         \_xeCJK_punct_if_right:NTF #2
5307         { \_xeCJK_save_kerning:nnNN { bound_kern } { bound } }
5308         { \_xeCJK_save_kerning:nnnNN { bound_kern } { bound } { offset } }
```

```

5309     }
5310     #1 #2
5311 }
5312 \cs_new:Npn \__xeCJK_punct_kerning_process_aux:NN #1#2
5313 {
5314     \bool_if:NTF \l__xeCJK_enabled_kerning_bool
5315     { \__xeCJK_calc_kerning_margin:NN #1 #2 }
5316     { \l__xeCJK_margin_dim }
5317 }
5318 \dim_new:N \l__xeCJK_minimum_bound_dim
5319 \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。

```

5320 \cs_new_protected:Npn \__xeCJK_save_kerning:nnNN #1#2
5321 { \__xeCJK_save_kerning:nnnNN {#1} {#2} {#2} }
5322 \cs_new_protected:Npn \__xeCJK_save_kerning:nnnNN #1#2#3#4#5
5323 {
5324     \dim_set:Nn \l__xeCJK_tmp_dim
5325     {
5326         \l__xeCJK_kerning_margin_dim
5327         - ( \__xeCJK_use_punct_dim:nNN {#2} \c__xeCJK_right_tl #4 )
5328         - ( \__xeCJK_use_punct_dim:nNN {#3} \c__xeCJK_left_tl #5 )
5329     }
5330     \__xeCJK_save_punct_dim:nNnn {#1} #4 #5 { \l__xeCJK_tmp_dim }
5331     \__xeCJK_save_punct_skip:nNnnnn {#1} #4 #5
5332     { \l__xeCJK_tmp_dim }
5333     { \l__xeCJK_margin_dim - \l__xeCJK_kerning_margin_dim }
5334     { \l__xeCJK_kerning_margin_dim - \l__xeCJK_minimum_bound_dim }
5335 }

```

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

```

5336 \cs_new:Npn \__xeCJK_original_kerning_margin:NN #1#2
5337 {
5338     \dim_eval:n
5339     {
5340         \__xeCJK_use_punct_dim:nNN
5341         { \__xeCJK_punct_if_right:NTF #1 { margin } { bound } } \c__xeCJK_right_tl #1
5342         +
5343         \__xeCJK_use_punct_dim:nNN
5344         { \__xeCJK_punct_if_right:NTF #2 { bound } { margin } } \c__xeCJK_left_tl #2
5345     }
5346 }

```

```

\__xeCJK_calc_kerning_margin:NN 5347 \cs_new:Npn \__xeCJK_calc_kerning_margin:NN #1#2
\__xeCJK_calc_kerning_margin_aux:NN 5348 {
5349     \dim_max:nn
5350     { \l__xeCJK_kerning_margin_minimum_dim }
5351     {
5352         \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
5353         { \l__xeCJK_minimum_bound_dim }
5354         {
5355             \bool_if:NTF \l__xeCJK_optimize_kerning_bool
5356             { \dim_max:nn { \l__xeCJK_minimum_bound_dim } }
5357             { \use:n }
5358             { \__xeCJK_calc_kerning_margin_aux:NN #1 #2 }
5359         }
5360     }
5361 }
5362 \cs_new:Npn \__xeCJK_calc_kerning_margin_aux:NN #1#2
5363 {
5364     \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
5365     { \__xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
5366     {
5367         \fp_if_nan:nTF { \l__xeCJK_kerning_total_ratio_fp }
5368         {
5369             \xeCJK_if_same_class:NNTF #1 #2

```

```

5370         { \_xeCJK_kerning_width_or_ratio:nNN { same } }
5371         { \_xeCJK_kerning_width_or_ratio:nNN { different } }
5372     }
5373     {
5374         \_xeCJK_calc_kerning_margin:nNN
5375         {
5376             \_xeCJK_dim_ratio:Nn \l__xeCJK_kerning_total_ratio_fp
5377             {
5378                 \_xeCJK_use_punct_dim:nN { width } #1 +
5379                 \_xeCJK_use_punct_dim:nN { width } #2
5380             }
5381         }
5382     }
5383 }
5384 #1 #2
5385 }

\_xeCJK_kerning_width_or_ratio:nNN 5386 \cs_new:Npn \_xeCJK_kerning_width_or_ratio:nNN #1#2#3
5387 {
5388     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
5389     { \use:c { l__xeCJK_#1_align_margin_dim } }
5390     {
5391         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_align_ratio_fp } }
5392         {
5393             \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
5394             { \l__xeCJK_kerning_margin_width_dim \use:none:n }
5395             { \_xeCJK_dim_ratio:Nn \l__xeCJK_kerning_ratio_fp }
5396         }
5397         { \_xeCJK_dim_ratio:cn { l__xeCJK_#1_align_ratio_fp } }
5398         { \l__xeCJK_margin_dim }
5399     }
5400 }

\_xeCJK_punct_min_bound:NN 5401 \cs_new:Npn \_xeCJK_punct_min_bound:NN #1#2
5402 {
5403     \dim_max:nn
5404     {
5405         \dim_min:nn
5406         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 }
5407         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 }
5408     }
5409     {
5410         \dim_min:nn
5411         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #2 }
5412         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #2 }
5413     }
5414 }

```

#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。

```

5415 \cs_new:Npn \_xeCJK_calc_kerning_margin:nNN #1#2#3
5416 {
5417     \dim_eval:n
5418     {
5419         (#1)
5420         - ( \_xeCJK_use_punct_dim:nNN
5421             { \_xeCJK_punct_if_right:NTF #2 { bound } { margin } }
5422             \c__xeCJK_left_tl #2 )
5423         - ( \_xeCJK_use_punct_dim:nNN
5424             { \_xeCJK_punct_if_right:NTF #3 { margin } { bound } }
5425             \c__xeCJK_right_tl #3 )
5426         - ( \_xeCJK_use_punct_dim:nN { dimen } #2 )
5427         - ( \_xeCJK_use_punct_dim:nN { dimen } #3 )
5428     }
5429 }

```

计算标点的左右实际边距和实际尺寸。

```

5430 \cs_new_protected:Npn \xeCJK_calc_punct_dimen:N #1

```

```

5431 {
5432   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl #1
5433   { \xeCJK_glyph_bounds:NN 1 #1 }
5434   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl #1
5435   { \xeCJK_glyph_bounds:NN 3 #1 }
5436   \__xeCJK_save_punct_dim:nNn { width } #1
5437   { \tex_fontcharwd:D \tex_font:D `#1 }
5438   \__xeCJK_save_punct_dim:nNn { dimen } #1
5439   {
5440     ( \__xeCJK_use_punct_dim:nN { width } #1 ) -
5441     ( \__xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 ) -
5442     ( \__xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 )
5443   }
5444 }

```

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

```

5445 \cs_new:Npn \xeCJK_glyph_bounds:NN #1#2
5446 { \tex_XeTeXglyphbounds:D #1 ~ \tex_XeTeXcharglyph:D `#2 \exp_stop_f: }

```

PunctStyle

```

5447 \keys_define:nn { xeCJK / options }
5448 { PunctStyle .code:n = \exp_args:Ne \__xeCJK_set_punct_style:n {#1} }
5449 \cs_new_protected:Npn \__xeCJK_set_punct_style:n #1
5450 {
5451   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5452   { \tl_set:Nn \l_xeCJK_punct_style_tl {#1} }
5453   {
5454     \prop_get:NnNF \c__xeCJK_punct_style_alias_prop
5455     {#1} \l_xeCJK_punct_style_tl
5456     { \__xeCJK_error:ne { punct-style-unknown } {#1} }
5457   }
5458 }
5459 \prop_const_from_keyval:Nn \c__xeCJK_punct_style_alias_prop
5460 {
5461   halfwidth      = banjiao ,
5462   fullwidth      = quanjiao ,
5463   mixedwidth     = kaiming ,
5464   marginkerning  = hangmobanjiao ,
5465   plain          = plain
5466 }
5467 \tl_new:N \l_xeCJK_punct_style_tl
5468 \tl_const:Nn \c__xeCJK_punct_style_plain_tl { plain }
5469 \__xeCJK_msg_new:nn { punct-style-unknown }
5470 {
5471   Punctuation~style~"#1"~is~unknown. \\\
5472   The~available~styles~are~listed~as~follow.\\
5473   "plain,~\seq_use:Nnnn \g__xeCJK_punct_style_seq
5474   { ~and~ } { ,~ } { ,~and~ }".\\
5475 }

```

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

```

5476 \cs_new_protected:Npn \__xeCJK_trim_spaces:n #1
5477 {
5478   \tl_set:Ne \ProcessedArgument
5479   { \exp_args:Ne \tl_trim_spaces:n {#1} }
5480 }

```

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

```

5481 \NewDocumentCommand \xeCJKDeclarePunctStyle
5482 { > { \__xeCJK_trim_spaces:n } m m }
5483 {
5484   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5485   { \__xeCJK_warning:ne { punct-style-already-defined } {#1} }
5486   { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
5487   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
5488 }

```

```

5489 \seq_new:N \g__xeCJK_punct_style_seq
5490 \__xeCJK_msg_new:nn { punct-style-already-defined }
5491 {
5492   Punctuation~style~"#1"~is~already~defined!. \\\
5493   The~existing~style~of~"#1"~will~be~overwritten.\\
5494 }
5495 \@onlypreamble \xeCJKDeclarePunctStyle

```

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

```

5496 \NewDocumentCommand \xeCJKEditPunctStyle
5497 { > { \__xeCJK_trim_spaces:n } m m }
5498 {
5499   \IfInstanceExistTF { xeCJK / punctuation } {#1}
5500   { \EditInstance { xeCJK / punctuation } {#1} {#2} }
5501   { \__xeCJK_error:ne { punct-style-unknown } {#1} }
5502 }
5503 \@onlypreamble \xeCJKEditPunctStyle

```

默认设置即为全角格式。

```

5504 \xeCJKDeclarePunctStyle { quanjiao }
5505 {
5506   optimize-kerning = true ,
5507   enabled-left-right-kerning = false
5508 }
5509 \xeCJKDeclarePunctStyle { hangmobanjiang } { enabled-kerning = false }
5510 \xeCJKDeclarePunctStyle { banjiao }
5511 {
5512   fixed-punct-ratio = 0.5 ,
5513   optimize-margin = true ,
5514   kerning-total-ratio = 0.5 ,
5515   optimize-kerning = true
5516 }
5517 \xeCJKDeclarePunctStyle { kaiming }
5518 {
5519   fixed-punct-ratio = 0.5 ,
5520   mixed-punct-ratio = 1.0 ,
5521   optimize-margin = true ,
5522   kerning-total-ratio = 0.5 ,
5523   optimize-kerning = true
5524 }
5525 \xeCJKDeclarePunctStyle { CCT }
5526 {
5527   fixed-punct-ratio = 0.7 ,
5528   optimize-margin = true ,
5529   kerning-total-ratio = 0.6 ,
5530   optimize-kerning = true
5531 }

```

5.11 后备字体

`AutoFallBack` 后备字体的宏包选项声明。

```

5532 \keys_define:nn { xeCJK / options }
5533 {
5534   AutoFallBack .choice: ,
5535   AutoFallBack / true .code:n =
5536   {
5537     \cs_set_eq:NN \xeCJK_fallback_symbol:NN
5538     \__xeCJK_fallback_symbol:NN
5539     \cs_set_eq:NN \xeCJK_fallback_punct_symbol:NN
5540     \__xeCJK_fallback_punct_symbol:NN
5541     \cs_set_eq:NN \xeCJK_clear_fallback_font:
5542     \__xeCJK_clear_fallback_font:

```

```

5543     } ,
5544     AutoFallBack / false .code:n =
5545     {
5546         \xeCJK_cs_clear:N \xeCJK_fallback_symbol:NN
5547         \xeCJK_cs_clear:N \xeCJK_fallback_punct_symbol:NN
5548         \xeCJK_cs_clear:N \xeCJK_clear_fallback_font:
5549     } ,
5550     AutoFallBack .default:n = { true } ,
5551     fallback .meta:n = { AutoFallBack = true }
5552 }

```

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
5553 \cs_new_protected:Npn \__xeCJK_fallback_symbol:NN #1#2
5554 {
5555     \xeCJK_reset_fallback_font:
5556     \xeCJK_glyph_if_exist:NF #2
5557     { \__xeCJK_fallback_symbol_aux:NN }
5558     #1#2
5559 }
5560 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol:NN #1#2
5561 {
5562     \xeCJK_reset_fallback_font:
5563     \xeCJK_glyph_if_exist:NF #2
5564     { \__xeCJK_fallback_punct_symbol_aux:NN }
5565     #1#2
5566 }
5567 \cs_new_eq:NN \xeCJK_fallback_symbol:NN \prg_do_nothing:
5568 \cs_new_eq:NN \xeCJK_fallback_punct_symbol:NN \prg_do_nothing:
5569 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:NN
5570 {
5571     \__xeCJK_fallback_symbol_aux:nnNN
5572     { \CJK@family }
5573     { \l_xeCJK_family_tl }
5574 }
5575 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol_aux:NN
5576 {
5577     \__xeCJK_fallback_symbol_aux:nnNN
5578     { \CJK@punctfamily }
5579     { \l_xeCJK_punct_family_tl }
5580 }
5581 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:nnNN
5582 {
5583     \cs_set_protected:Npe \xeCJK_reset_fallback_font:
5584     {
5585         \tex_the:D \tex_font:D
5586         \xeCJK_clear_fallback_font:
5587     }
5588     \exp_args:Nee \__xeCJK_fallback_loop:nnNN
5589 }
5590 \cs_new_protected:Npn \__xeCJK_clear_fallback_font:
5591 { \cs_set_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing: }
5592 \cs_new_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing:
5593 \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。

```

5594 \cs_new_protected:Npn \__xeCJK_fallback_loop:nnNN
5595 {
5596     \cs_set_eq:NN \__xeCJK_fallback_loop:TF \use_i:nn
5597     \__xeCJK_fallback_loop:nnnNN { FallBack }
5598 }
5599 \cs_new_protected:Npn \__xeCJK_fallback_loop:nnnNN #1#2#3
5600 {
5601     \xeCJK_select_fallback_font:nnn {#1} {#2} {#3}
5602     \__xeCJK_fallback_loop:TF

```

```

5603     { \_xeCJK_fallback_loop_aux:nnnNN }
5604     { \_xeCJK_fallback_missing_glyph:nnnNN }
5605     {#1} {#2} {#3}
5606 }
5607 \cs_new_protected:Npn \_xeCJK_fallback_loop_aux:nnnNN #1#2#3#4#5
5608 {
5609     \xeCJK_glyph_if_exist:NF #5
5610     { \_xeCJK_fallback_loop:nnnNN { #1/FallBack } {#2} {#3} }
5611     #4#5
5612 }
5613 \cs_new_protected:Npn \_xeCJK_fallback_missing_glyph:nnnNN #1#2#3#4#5
5614 {
5615     \_xeCJK_warning:neee { missing-glyph } {#1} {#2} {#5}
5616     #4#5
5617 }
5618 \cs_new_protected:Npn \xeCJK_select_fallback_font:nnn #1#2
5619 {
5620     \_xeCJK_select_fallback_font:cnnn
5621     { \_xeCJK_font_csname:n { #2/#1 } } {#1} {#2}
5622 }
5623 \cs_new_protected:Npn \_xeCJK_select_fallback_font:Nnnn #1
5624 {
5625     \cs_if_exist:NF #1
5626     { \_xeCJK_fallback_font_initial:NNnnn }
5627     #1 \use_none:nnn
5628 }
5629 \cs_generate_variant:Nn \_xeCJK_select_fallback_font:Nnnn { c }
5630 \cs_new_protected:Npn \_xeCJK_fallback_font_initial:NNnnn #1#2#3#4#5
5631 {
5632     \xeCJK_family_if_exist:nTF { #5/#3 }
5633     { \_xeCJK_font_initial:Nn #1 { #5/#3 } }
5634     { \_xeCJK_fallback_font_initial_auxi:Nnnn #1 {#5} {#3} {#4} }
5635     #1
5636 }
5637 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxi:Nnnn #1
5638 {
5639     \exp_args:NNe \_xeCJK_fallback_font_initial_auxii:Nnnnn
5640     #1 { \CJKfamilydefault }
5641 }
5642 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxii:Nnnnn #1#2#3
5643 {
5644     \str_if_eq:nnTF {#2} {#3}
5645     { \_xeCJK_fallback_loop_end:Nnnn }
5646     { \_xeCJK_fallback_font_initial_auxiii:Nnnn }
5647     #1 {#2}
5648 }
5649 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxiii:Nnnn #1#2
5650 {
5651     \xeCJK_family_if_exist:nTF {#2}
5652     { \_xeCJK_fallback_font_initial_auxiv:Nnnn }
5653     { \_xeCJK_fallback_loop_end:Nnnn }
5654     #1 {#2}
5655 }
5656 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxiv:Nnnn #1#2#3#4
5657 {
5658     \_xeCJK_font_initial:Nn #1 {#2}
5659     \exp_args:Nc \_xeCJK_fallback_font_initial_auxiii:Nnnn
5660     { \_xeCJK_font_csname:n { #4/#3/FallBack } }
5661     { #2/FallBack } { #3/FallBack } {#4}
5662 }
5663 \cs_new_eq:NN \_xeCJK_fallback_loop:TF \use_i:nn
5664 \cs_new_protected:Npn \_xeCJK_fallback_loop_end:Nnnn #1#2#3#4
5665 { \cs_gset_eq:NN #1 \_xeCJK_fallback_loop_end: }
5666 \cs_new_protected:Npn \_xeCJK_fallback_loop_end:
5667 { \cs_set_eq:NN \_xeCJK_fallback_loop:TF \use_ii:nn }
5668 \_xeCJK_msg_new:nn { missing-glyph }
5669 {

```

```

5670     CJKfamily~\__xeCJK_msg_family_map:n {#2}'~(#{1})~
5671     does~not~contain~glyph~`#3'~
5672     ( U + \int_to_Hex:n { `#3 } )~\msg_line_context:.
5673 }

\setCJKfallbackfamilyfont 5674 \NewDocumentCommand \setCJKfallbackfamilyfont { m o m }
5675 {
5676     \__xeCJK_pass_args:nnnn
5677     { \xeCJK_set_family_fallback:nnn {#1} } {#2} {#3}
5678     { }
5679 }

xeCJK_set_family_fallback:nnn 5680 \cs_new_protected:Npn \xeCJK_set_family_fallback:nnn #1#2#3
5681 {
5682     \group_begin:
5683     \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
5684     \prop_get:NoNF \g__xeCJK_family_font_name_prop
5685     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
5686     { \tl_clear:N \l__xeCJK_font_name_tl }
5687     \clist_set:Nn \l__xeCJK_public_options_clist {#2}
5688     \clist_map_function:nN {#3} \__xeCJK_set_family_fallback:n
5689     \group_end:
5690 }
5691 \cs_new_protected:Npn \__xeCJK_set_family_fallback:n #1
5692 {
5693     \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /Fallback }
5694     \__xeCJK_get_sub_features:on \l__xeCJK_fallback_family_tl {#1}
5695     \clist_concat:NNN \l__xeCJK_sub_font_options_clist
5696     \l__xeCJK_public_options_clist
5697     \l__xeCJK_sub_font_options_clist
5698     \exp_args:Nooo \xeCJK_set_family:nnn
5699     \l__xeCJK_fallback_family_tl
5700     \l__xeCJK_sub_font_options_clist
5701     \l__xeCJK_sub_font_name_tl
5702 }
5703 \tl_new:N \l__xeCJK_fallback_family_tl
5704 \clist_new:N \l__xeCJK_public_options_clist

```

5.12 CJK 字体族声明方式

```

5705 \bool_new:N \g__xeCJK_auto_fake_bold_bool
5706 \bool_new:N \g__xeCJK_auto_fake_slant_bool
5707 \fp_new:N \g__xeCJK_embolden_factor_fp
5708 \fp_new:N \g__xeCJK_slant_factor_fp

```

AutoFakeBold 伪粗体和伪斜体的宏包选项声明。

```

AutoFakeSlant
EmboldenFactor
SlantFactor
5709 \keys_define:nn { xeCJK / options }
5710 {
5711     AutoFakeBold .choices:nn = { true , false }
5712     { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_bold_bool } ,
5713     AutoFakeBold / unknown .code:n =
5714     {
5715         \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool
5716         \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
5717     } ,
5718     AutoFakeSlant .default:n = { true } ,
5719     AutoFakeSlant .choices:nn = { true , false }
5720     { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_slant_bool } ,
5721     AutoFakeSlant / unknown .code:n =
5722     {
5723         \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
5724         \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
5725     } ,
5726     AutoFakeSlant .default:n = { true } ,
5727     EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
5728     SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
5729     BoldFont .meta:n = { AutoFakeBold = true } ,

```

```

5730     boldfont .meta:n = { AutoFakeBold = true } ,
5731     SlantFont .meta:n = { AutoFakeSlant = true } ,
5732     slantfont .meta:n = { AutoFakeSlant = true }
5733 }

```

\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

```

5734 \seq_new:N \g__xeCJK_sub_key_seq
5735 \cs_new_protected:Npn \xeCJK_new_sub_key:n #1
5736 {
5737     \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
5738     \keys_define:nn { xeCJK / features }
5739     {
5740         #1 .code:n =
5741         {
5742             \tl_if_blank:nTF {##1}
5743             {
5744                 \prop_clear:N \l__xeCJK_sub_key_prop
5745                 \tl_set:Nx \l__xeCJK_sub_family_name_tl
5746                 { \l__xeCJK_family_name_tl /#1 }
5747                 \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
5748             }
5749             {
5750                 \tl_clear:N \l__xeCJK_sub_family_name_tl
5751                 \str_if_eq:nnTF {##1} { * }
5752                 { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
5753                 { \__xeCJK_get_sub_features:nn {#1} {##1} }
5754             }
5755         } ,
5756         #1 .default:n = { }
5757     }
5758 }

```

__xeCJK_get_sub_features:nn

__xeCJK_get_sub_features:w

```

5759 \cs_new_protected:Npn \__xeCJK_get_sub_features:nn #1#2
5760 {
5761     \tl_set:Nx \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
5762     \clist_clear:N \l__xeCJK_sub_font_options_clist
5763     \exp_after:wN \__xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
5764     \q_mark [ \q_nil ] \q_mark \q_stop
5765     \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
5766     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
5767     { \tl_replace_all:Nno \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
5768     \prop_put:Nne \l__xeCJK_sub_key_prop {#1}
5769     {
5770         { \exp_not:o \l__xeCJK_sub_font_options_clist }
5771         { \exp_not:o \l__xeCJK_sub_font_name_tl }
5772     }
5773 }
5774 \cs_new_protected:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
5775 {
5776     \quark_if_nil:nTF {#2}
5777     { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
5778     {
5779         \tl_set:Nx \l__xeCJK_sub_font_name_tl
5780         { \xeCJK_tl_remove_outer_braces:n {#3} }
5781         \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
5782         { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
5783         { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
5784     }
5785 }
5786 \tl_new:N \l__xeCJK_sub_family_name_tl
5787 \tl_new:N \l__xeCJK_sub_font_name_tl
5788 \clist_new:N \l__xeCJK_sub_font_options_clist
5789 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { o }
5790 \cs_generate_variant:Nn \tl_replace_all:Nnn { Nno }

```

FallBack 5791 \xeCJK_new_sub_key:n { FallBack }

调用字体的属性声明, 同 `fontspec` 宏包。

`BoldFont`
`ItalicFont`

```
5792 \keys_define:nn { xeCJK / features }
5793 {
5794   BoldFont .tl_set:N = \l__xeCJK_font_name_bf_tl ,
5795   ItalicFont .tl_set:N = \l__xeCJK_font_name_it_tl
5796 }
```

`AutoFakeBold`
`AutoFakeSlant`

```
5797 \keys_define:nn { xeCJK / features }
5798 {
5799   AutoFakeBold .choice: ,
5800   AutoFakeBold / true .code:n =
5801   {
5802     \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
5803     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
5804   } ,
5805   AutoFakeBold / false .code:n =
5806   { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
5807   AutoFakeBold / unknown .code:n =
5808   {
5809     \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
5810     \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
5811   } ,
5812   AutoFakeBold .default:n = { true } ,
5813   AutoFakeSlant .choice: ,
5814   AutoFakeSlant / true .code:n =
5815   {
5816     \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
5817     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
5818   } ,
5819   AutoFakeSlant / false .code:n =
5820   { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
5821   AutoFakeSlant / unknown .code:n =
5822   {
5823     \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
5824     \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
5825   } ,
5826   AutoFakeSlant .default:n = { true }
5827 }
```

`\__xeCJK_set_family_initial:`

```
5828 \cs_new_protected:Npn \__xeCJK_set_family_initial:
5829 {
5830   \int_gincr:N \g__xeCJK_family_int
5831   \prop_clear:N \l__xeCJK_sub_key_prop
5832   \tl_clear:N \l__xeCJK_font_name_bf_tl
5833   \tl_clear:N \l__xeCJK_font_name_it_tl
5834   \tl_clear:N \l__xeCJK_sub_family_name_tl
5835   \clist_clear:N \l__xeCJK_fontspec_options_clist
5836   \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
5837   \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
5838   \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
5839   \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
5840 }
5841 \int_new:N \g__xeCJK_family_int
5842 \prop_new:N \l__xeCJK_sub_key_prop
5843 \clist_new:N \l__xeCJK_fontspec_options_clist
5844 \bool_new:N \l__xeCJK_auto_fake_bold_bool
5845 \bool_new:N \l__xeCJK_auto_fake_slant_bool
5846 \fp_new:N \l__xeCJK_embolden_factor_fp
5847 \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。

```
5848 \cs_new_protected:Npn \xeCJK_set_family:nnn #1#2#3
5849 {
5850   \group_begin:
5851   \__xeCJK_set_family_initial:
5852   \tl_set:Nn \l__xeCJK_family_name_tl {#1}
```

```

5853 \clist_set:Nn \l__xeCJK_font_options_clist {#2}
5854 \tl_set:Nn \l__xeCJK_font_name_tl {#3}
5855 \clist_concat:NNN \l__xeCJK_font_options_clist
5856 \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
5857 \keys_set_known:noN { xeCJK / features }
5858 \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
5859 \__xeCJK_binding_sub_family:
5860 \__xeCJK_parse_font_shape:
5861 \__xeCJK_check_family:o \l__xeCJK_family_name_tl
5862 \__xeCJK_gset_family_cs:n { \l__xeCJK_family_name_tl }
5863 \__xeCJK_save_family_info:
5864 \__xeCJK_set_sub_block_family:
5865 \group_end:
5866 }
5867 \tl_new:N \l__xeCJK_family_name_tl
5868 \tl_new:N \l__xeCJK_font_name_tl
5869 \clist_new:N \l__xeCJK_font_options_clist
5870 \cs_generate_variant:Nn \xeCJK_set_family:nnn { e , o }

```

```

\__xeCJK_binding_sub_family: 5871 \cs_new_protected:Npn \__xeCJK_binding_sub_family:
5872 {
5873 \tl_if_empty:NF \l__xeCJK_sub_family_name_tl
5874 { \tl_set_eq:NN \l__xeCJK_family_name_tl \l__xeCJK_sub_family_name_tl }
5875 }

```

```

\__xeCJK_gset_family_cs:n 5876 \cs_new_protected:Npn \__xeCJK_gset_family_cs:n #1
5877 {
5878 \cs_gset_protected:cpe { \__xeCJK_family_csname:n {#1} }
5879 {
5880 \group_begin:
5881 \exp_not:n { \cs_set_eq:NN \xeCJK@fontfamily \use_none:n }
5882 \exp_not:n { \fontspec_gset_family:Nnn \g__xeCJK_fontspec_family_tl }
5883 { \exp_not:o \l__xeCJK_fontspec_options_clist }
5884 { \exp_not:o \l__xeCJK_font_name_tl }
5885 \__xeCJK_gset_family_nfss_cs:no
5886 {#1} { \exp_not:N \g__xeCJK_fontspec_family_tl }
5887 \group_end:
5888 \tl_set_eq:NN \exp_not:N \l__xeCJK_fontspec_family_tl
5889 \exp_not:N \g__xeCJK_fontspec_family_tl
5890 }
5891 }
5892 \tl_new:N \g__xeCJK_fontspec_family_tl
5893 \tl_new:N \l__xeCJK_fontspec_family_tl

```

```

\__xeCJK_check_family:n 5894 \cs_new_protected:Npn \__xeCJK_check_family:n #1
5895 {
5896 \prop_gpop:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
5897 {
5898 \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
5899 {
5900 \cs_undefine:c { \__xeCJK_family_csname:n {#1} }
5901 \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} }
5902 }
5903 \__xeCJK_warning:nee { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
5904 }
5905 }
5906 \cs_generate_variant:Nn \__xeCJK_check_family:n { o }
5907 \__xeCJK_msg_new:nn { CJKfamily-redef }
5908 { Redefining~CJKfamily~\__xeCJK_msg_family_map:n {#1}'~(#2). }

```

```

\__xeCJK_parse_font_shape: 5909 \cs_new_protected:Npn \__xeCJK_parse_font_shape:
5910 {
5911 \tl_if_blank:oTF \l__xeCJK_font_name_bf_tl
5912 {
5913 \bool_if:NT \l__xeCJK_auto_fake_bold_bool
5914 {
5915 \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5916 { AutoFakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }

```

```

5917     }
5918   }
5919   {
5920     \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5921     { BoldFont = { \exp_not:o \l__xeCJK_font_name_bf_tl } }
5922   }
5923   \tl_if_blank:oTF \l__xeCJK_font_name_it_tl
5924   {
5925     \bool_if:NT \l__xeCJK_auto_fake_slant_bool
5926     {
5927       \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5928       { AutoFakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
5929     }
5930   }
5931   {
5932     \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
5933     { ItalicFont = { \exp_not:o \l__xeCJK_font_name_it_tl } }
5934   }
5935 }

```

```

\g__xeCJK_family_name_prop 5936 \prop_new:N \g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop 5937 \prop_new:N \g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop 5938 \prop_new:N \g__xeCJK_family_font_options_prop

```

```

\__xeCJK_save_family_info: 5939 \cs_new_protected:Npn \__xeCJK_save_family_info:
5940 {
5941   \exp_args:Nooo \__xeCJK_save_family_info:nnn
5942   \l__xeCJK_family_name_tl
5943   \l__xeCJK_font_name_tl
5944   \l__xeCJK_font_options_clist
5945 }
5946 \cs_new_protected:Npn \__xeCJK_save_family_info:nnn #1#2#3
5947 {
5948   \prop_gput:Nnn \g__xeCJK_family_font_name_prop {#1} {#2}
5949   \prop_gput:Nnn \g__xeCJK_family_font_options_prop {#1} {#3}
5950 }

```

```

\__xeCJK_set_sub_block_family: 5951 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:
5952 {
5953   \prop_if_empty:NF \l__xeCJK_sub_key_prop
5954   {
5955     \prop_map_function:NN
5956     \l__xeCJK_sub_key_prop
5957     \__xeCJK_set_sub_block_family:nn
5958   }
5959 }
5960 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:nn #1#2
5961 {
5962   \tl_set:Ne \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/#1 }
5963   \quark_if_no_value:nTF {#2}
5964   { \__xeCJK_copy_sub_family:n {#1} }
5965   { \xeCJK_set_family:onn \l__xeCJK_sub_family_name_tl #2 }
5966 }
5967 \cs_new_protected:Npn \__xeCJK_copy_sub_family:n #1
5968 {
5969   \__xeCJK_check_family:o \l__xeCJK_sub_family_name_tl
5970   \prop_get:NoNT \g__xeCJK_family_font_name_prop
5971   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
5972   {
5973     \prop_gput:Noo \g__xeCJK_family_font_name_prop
5974     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
5975   }
5976   \prop_get:NoNT \g__xeCJK_family_font_options_prop
5977   \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
5978   {
5979     \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
5980     \prop_gput:Noo \g__xeCJK_family_font_options_prop
5981     \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist

```

```

5982     }
5983     \cs_gset_protected:cpe
5984     { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
5985     {
5986         \xeCJK_family_if_exist:et { \l__xeCJK_family_name_tl }
5987         {
5988             \__xeCJK_gset_family_nfss_cs:no
5989             { \l__xeCJK_sub_family_name_tl }
5990             { \exp_not:N \l__xeCJK_fontspec_family_tl }
5991         }
5992     }
5993 }

\__xeCJK_copy_family:nn 5994 \cs_new_protected:Npn \__xeCJK_copy_family:nn #1#2
\__xeCJK_copy_family:ee 5995 {
5996     \xeCJK_family_if_exist:nT {#2}
5997     {
5998         \prop_gput:Nno \g__xeCJK_family_name_prop
5999         {#1} \l__xeCJK_fontspec_family_tl
6000         \tl_map_inline:nn
6001         {
6002             \g__xeCJK_family_font_name_prop
6003             \g__xeCJK_family_font_options_prop
6004         }
6005         {
6006             \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
6007             { \prop_gput:Nno ##1 {#1} \l__xeCJK_tmp_tl }
6008         }
6009         \cs_gset_eq:cc
6010         { \__xeCJK_family_nfss_csname:n {#1} }
6011         { \__xeCJK_family_nfss_csname:n {#2} }
6012     }
6013 }
6014 \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 6015 \cs_new:Npn \__xeCJK_font_csname:n #1
6016 { xeCJK/#1/\f@series/\f@shape/\f@size }
6017 \tl_new:N \l_xeCJK_current_font_tl
6018 \tl_set:N \l_xeCJK_current_font_tl
6019 { \__xeCJK_font_csname:n { \CJK@family } }
6020 \cs_new_protected:Npn \xeCJK_select_font:
6021 {
6022     \__xeCJK_select_font:cn
6023     { \l_xeCJK_current_font_tl }
6024     { \l_xeCJK_family_tl }
6025 }
6026 \cs_new_protected:Npn \__xeCJK_select_font:Nn #1#2
6027 {
6028     \xeCJK_clear_fallback_font:
6029     \cs_if_exist:NF #1 { \__xeCJK_font_initial:Nn #1 {#2} }
6030     #1
6031 }
6032 \cs_generate_variant:Nn \__xeCJK_select_font:Nn { c }
6033 \tl_new:N \l__xeCJK_current_coord_tl
6034 \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)。

6035 \cs_new_protected:Npn \__xeCJK_font_initial:Nn #1#2
6036 {
6037     \group_begin:

```

```

6038     \__xeCJK_font_initial_hook:
6039     \__xeCJK_family_use:n {#2}
6040     \xeCJK_font_gset_to_current:N #1
6041     \group_end:
6042   }
6043   \cs_new_protected:Npn \__xeCJK_font_initial_hook:
6044   { \tl_use:N \g__xeCJK_font_initial_hook_tl }
6045   \cs_new_protected:Npn \__xeCJK_gadd_font_initial_hook:n
6046   { \tl_gput_right:Nn \g__xeCJK_font_initial_hook_tl }
6047   \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
6048 \cs_new_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
6049 \cs_new_protected:Npn \__xeCJK_select_punct_font_aux:
6050 {
6051   \__xeCJK_select_font:cn
6052   { \l_xeCJK_current_punct_font_tl }
6053   { \l_xeCJK_punct_family_tl }
6054 }
6055 \tl_new:N \CJK@punctfamily
6056 \tl_new:N \l_xeCJK_punct_family_tl
6057 \tl_new:N \l_xeCJK_current_punct_font_tl
6058 \tl_set:Nn \CJK@punctfamily { \CJK@family }
6059 \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
6060 \tl_set:Nn \l_xeCJK_current_punct_font_tl
6061 { \__xeCJK_font_csname:n { \CJK@punctfamily } }
6062 \cs_new_eq:NN \__xeCJK_select_font: \prg_do_nothing:
6063 \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
6064 \cs_new_protected:Npn \__xeCJK_switch_font:nn #1#2
6065 {
6066   \str_if_eq:nnF {#1} {#2}
6067   {
6068     \__xeCJK_info:nee { CJK-block } {#1} {#2}
6069     \str_if_eq:nnTF {#2} { CJK }
6070     { \xeCJK_select_font: }
6071     { \xeCJK_select_font:n {#2} }
6072   }
6073 }
6074 \__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。

```

6075 \cs_new_protected:Npn \xeCJK_select_font:n #1
6076 {
6077   \__xeCJK_select_font:cnn
6078   { \__xeCJK_font_csname:n { \CJK@family/#1 } }
6079   { \l_xeCJK_family_tl }
6080   {#1}
6081 }
6082 \cs_new_protected:Npn \__xeCJK_select_font:Nnn #1#2#3
6083 {
6084   \xeCJK_clear_fallback_font:
6085   \cs_if_exist:NF #1
6086   { \__xeCJK_block_font_initial:Nnn #1 {#2} {#3} }
6087   #1
6088 }
6089 \cs_generate_variant:Nn \__xeCJK_select_font:Nnn { c }
6090 \cs_new_protected:Npn \__xeCJK_block_font_initial:Nnn #1#2#3
6091 {
6092   \xeCJK_block_family:nn {#2} {#3}
6093   \__xeCJK_font_initial:Nn #1 { #2/#3 }
6094 }

```

```

6095 \cs_new_protected:Npn \xeCJK_block_family:nn #1#2
6096 {
6097   \xeCJK_family_if_exist:ef { #1/#2 }
6098   {
6099     \__xeCJK_copy_family:ee { #1/#2 }
6100     {
6101       \cs_if_exist:cTF
6102       { \__xeCJK_family_csname:n { \CJKfamilydefault/#2 } }
6103       { \CJKfamilydefault/#2 } {#1}
6104     }
6105   }
6106 }

\__xeCJK_family_csname:n 6107 \cs_new:Npn \__xeCJK_family_csname:n #1
\__xeCJK_family_nfss_csname:n 6108 { xeCJK/family/#1 }
\__xeCJK_family_use:n 6109 \cs_new:Npn \__xeCJK_family_nfss_csname:n #1
\__xeCJK_gset_family_nfss_cs:nn 6110 { xeCJK/family/nfss/#1 }
6111 \cs_new_protected:Npn \__xeCJK_family_use:n #1
6112 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
6113 \cs_new_protected:Npn \__xeCJK_gset_family_nfss_cs:nn #1#2
6114 {
6115   \prop_gput:Nnn \g__xeCJK_family_name_prop {#1} {#2}
6116   \cs_gset_protected:cpe
6117   { \__xeCJK_family_nfss_csname:n {#1} }
6118   { \__xeCJK_nfss_family:nn { \c__xeCJK_encoding_tl } {#2} }
6119 }
6120 \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。

```

6121 \cs_if_exist:NTF \fontseriesforce
6122 {
6123   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
6124   {
6125     \fontencoding {#1}
6126     \str_if_eq:eeF { \f@series } { \bfdefault }
6127     {
6128       \str_case:e:nn { \f@family }
6129       {
6130         { \rmdefault } { \__xeCJK_nfss_series:n { rm } }
6131         { \sfdefault } { \__xeCJK_nfss_series:n { sf } }
6132         { \ttdefault } { \__xeCJK_nfss_series:n { tt } }
6133       }
6134     }
6135     \fontfamily {#2}
6136     \selectfont
6137   }
6138   \cs_new_protected:Npn \__xeCJK_nfss_series:n #1
6139   {
6140     \str_if_eq:eeT { \f@series } { \use:c { bfseries@#1 } }
6141     { \fontseriesforce { \bfdefault } }
6142   }
6143 }
6144 {
6145   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
6146   {
6147     \fontencoding {#1}
6148     \tl_set:Nn \f@family {#2}
6149     \selectfont
6150   }
6151 }

```

```

\xeCJK_family_if_exist:nTF 6152 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:n #1 { T , F , TF }
6153 {
6154   \prop_get:NnNTF \g__xeCJK_family_name_prop
6155   {#1} \l__xeCJK_fontspec_family_tl
6156   { \prg_return_true: }
6157   {

```

```

6158     \exp_args:Ne \cs_if_exist_use:cTF
6159     { \_xeCJK_family_csname:n {#1} }
6160     { \prg_return_true: }
6161     { \prg_return_false: }
6162   }
6163 }
6164 \prg_generate_conditional_variant:Nnn \xeCJK_family_if_exist:n { e } { T , F , TF }

```

\CJKfamily 用于切换 CJK 字体族。

```

6165 \NewDocumentCommand \CJKfamily { t+ t- m }
6166 {
6167   \xeCJK_family:NNe #1 #2 {#3}
6168   \tex_ignorespaces:D
6169 }
6170 \cs_new_protected:Npn \xeCJK_family:NNn #1#2#3
6171 {
6172   \tl_if_blank:nTF {#3}
6173   {
6174     \bool_if:NF #1 { \bool_if:NF #2 { \use_none:nn } }
6175     \xeCJK_family_if_exist_use:e { \l_xeCJK_family_tl }
6176   }
6177   {
6178     \bool_if:NTF #2
6179     { \xeCJK_family_if_exist_use:n {#3} }
6180     {
6181       \xeCJK_family_if_exist:nTF {#3}
6182       {
6183         \tl_set:Nn \l_xeCJK_family_tl {#3}
6184         \tl_set_eq:NN \CJK@family \l_xeCJK_fontspec_family_tl
6185         \bool_if:NT #1 { \_xeCJK_family_use:n {#3} }
6186       }
6187       { \_xeCJK_family_unknown_warning:n {#3} }
6188     }
6189   }
6190 }
6191 \cs_generate_variant:Nn \xeCJK_family:NNn { NNe }
6192 \cs_new_protected:Npn \xeCJK_switch_family:n #1
6193 {
6194   \xeCJK_family_if_exist:nTF {#1}
6195   {
6196     \tl_set:Nn \l_xeCJK_family_tl {#1}
6197     \tl_set_eq:NN \CJK@family \l_xeCJK_fontspec_family_tl
6198   }
6199   { \_xeCJK_family_unknown_warning:n {#1} }
6200 }
6201 \cs_generate_variant:Nn \xeCJK_switch_family:n { e , o }

```

PunctFamily 设置汉字标点符号的字体。

```

6202 \keys_define:nn { xeCJK / options }
6203 {
6204   PunctFamily .choice: ,
6205   PunctFamily .value_required:n = { true } ,
6206   PunctFamily / false .code:n =
6207   {
6208     \tl_set:Nn \CJK@punctfamily { \CJK@family }
6209     \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
6210     \xeCJK_cs_clear:N \_xeCJK_select_font:
6211     \xeCJK_cs_clear:N \_xeCJK_select_punct_font:
6212     \cs_set_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
6213   } ,
6214   PunctFamily / unknown .code:n =
6215   { \xeCJK_punct_family:e {#1} } ,
6216 }
6217 \cs_new_protected:Npn \xeCJK_punct_family:n #1
6218 {

```

```

6219 \xeCJK_family_if_exist:nTF {#1}
6220 {
6221     \tl_set:Nn \l_xeCJK_punct_family_tl {#1}
6222     \tl_set_eq:NN \CJK@punctfamily \l_xeCJK_fontspec_family_tl
6223     \cs_set_eq:NN \_xeCJK_select_font: \xeCJK_select_font:
6224     \cs_set_eq:NN \_xeCJK_select_punct_font: \_xeCJK_select_punct_font_aux:
6225     \cs_set_eq:NN \xeCJK_select_punct_font: \_xeCJK_select_punct_font:
6226 }
6227 { \_xeCJK_family_unknown_warning:n {#1} }
6228 }
6229 \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。

```
6230 \tl_new:N \l_xeCJK_family_tl
```

\CJK@family 用于保存实际的字体族名称。

```
6231 \tl_new:N \CJK@family
```

```

\__xeCJK_gobble_CJKfamily: 6232 \cs_new_protected:Npn \__xeCJK_gobble_CJKfamily:
6233 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
6234 \NewExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }

```

```

\xeCJK_family_if_exist_use:n 6235 \cs_new_protected:Npn \xeCJK_family_if_exist_use:n #1
6236 {
6237     \xeCJK_family_if_exist:nTF {#1}
6238     { \_xeCJK_family_use:n {#1} }
6239     { \_xeCJK_family_unknown_warning:n {#1} }
6240 }
6241 \cs_generate_variant:Nn \xeCJK_family_if_exist_use:n { e }

```

```

\__xeCJK_family_unknown_warning:n 6242 \cs_new_protected:Npn \__xeCJK_family_unknown_warning:n #1
6243 {
6244     \prop_if_empty:NF \g__xeCJK_family_font_name_prop
6245     {
6246         \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
6247         {
6248             \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
6249             \_xeCJK_warning:ne { CJKfamily-Unknown } {#1}
6250         }
6251     }
6252 }
6253 \seq_new:N \g__xeCJK_unknown_family_seq
6254 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
6255 {
6256     Unknown~CJK~family~~\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
6257     Try~to~use~~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
6258 }
6259 \cs_new:Npn \__xeCJK_msg_def_family_map:n #1
6260 {
6261     \str_case_e:nnF {#1}
6262     {
6263         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
6264         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
6265         \CJKttdefault { \token_to_str:N \setCJKmonofont }
6266     }
6267     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
6268     [<...>]\{<...>\}
6269 }
6270 \cs_new:Npn \__xeCJK_msg_family_map:n #1
6271 {
6272     \str_case_e:nnF {#1}
6273     {
6274         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
6275         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
6276         \CJKttdefault { \token_to_str:N \CJKttdefault }
6277     }
6278     {#1}
6279 }

```

`\_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。

```

6280 \cs_new_protected:Npn \_xeCJK_pass_args:nnnn #1#2#3#4
6281 {
6282   \tl_if_novalue:nTF {#2}
6283   { \_xeCJK_post_arg:w {#1} {#3} {#4} }
6284   {
6285     \use:e { #1 {#2} {#3} }
6286     #4
6287   }
6288 }
6289 \NewDocumentCommand \_xeCJK_post_arg:w { m m m O { } }
6290 {
6291   \use:e { #1 {#4} {#2} }
6292   #3
6293 }

```

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

```

\setCJKsansfont
\setCJKmonofont
6294 \NewDocumentCommand \setCJKmainfont { o m }
6295 {
6296   \_xeCJK_pass_args:nnnn
6297   { \xeCJK_set_family:nnn { \CJKrmdefault } } {#1} {#2}
6298   { \_xeCJK_preamble_family:n { \CJKrmdefault } }
6299 }
6300 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
6301 \NewDocumentCommand \setCJKsansfont { o m }
6302 {
6303   \_xeCJK_pass_args:nnnn
6304   { \xeCJK_set_family:nnn { \CJKsfdefault } } {#1} {#2}
6305   { \_xeCJK_preamble_family:n { \CJKsfdefault } }
6306 }
6307 \NewDocumentCommand \setCJKmonofont { o m }
6308 {
6309   \_xeCJK_pass_args:nnnn
6310   { \xeCJK_set_family:nnn { \CJKttdefault } } {#1} {#2}
6311   { \_xeCJK_preamble_family:n { \CJKttdefault } }
6312 }
6313 \@onlypreamble \setCJKmainfont
6314 \@onlypreamble \setCJKmathfont
6315 \@onlypreamble \setCJKsansfont
6316 \@onlypreamble \setCJKmonofont
6317 \@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。

```

6318 \cs_new_protected:Npn \_xeCJK_preamble_family:n #1
6319 { \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
6320 \NewDocumentCommand \setCJKfamilyfont { m o m }
6321 {
6322   \_xeCJK_pass_args:nnnn
6323   { \xeCJK_set_family:nnn {#1} } {#2} {#3}
6324   { }
6325 }
6326 \_xeCJK_msg_new:nn { command-already-defined }
6327 { Control~sequence~#1~is~already~defined. }
6328 \NewDocumentCommand \newCJKfontfamily { o m o m }
6329 {
6330   \tl_set:Nx \l__xeCJK_tmp_tl
6331   { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
6332   \cs_if_exist:NTF #2
6333   { \_xeCJK_error:ne { command-already-defined } { \token_to_str:N #2 } }
6334   {

```

```

6335         \cs_set_protected:Npe #2
6336         { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
6337     }
6338     \__xeCJK_pass_args:nnnn
6339     { \xeCJK_set_family:nnn { \l__xeCJK_tmp_tl } } {#3} {#4}
6340     { }
6341 }
6342 \NewDocumentCommand \CJKfontspec { o m }
6343 {
6344     \__xeCJK_pass_args:nnnn
6345     { \xeCJK_fontspec:nn } {#1} {#2}
6346     { \tex_ignorespaces:D }
6347 }

\xeCJK_fontspec:nn 6348 \cs_new_protected:Npn \xeCJK_fontspec:nn #1#2
6349 {
6350     \prop_get:NnNTF \g__xeCJK_fontspec_prop
6351     { CJKfontspec/#1/#2/id } \l_xeCJK_family_tl
6352     { \xeCJK_switch_family:o { \l_xeCJK_family_tl } }
6353     {
6354         \__xeCJK_fontspec:enn
6355         { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + 1 } ) }
6356         {#1} {#2}
6357     }
6358 }
6359 \cs_new_protected:Npn \__xeCJK_fontspec:nnn #1#2#3
6360 {
6361     \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
6362     \xeCJK_set_family:nnn {#1} {#2} {#3}
6363     \xeCJK_switch_family:n {#1}
6364 }
6365 \cs_generate_variant:Nn \xeCJK_fontspec:nn { oo }
6366 \cs_generate_variant:Nn \__xeCJK_fontspec:nnn { e }
6367 \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
6368 \clist_new:N \g__xeCJK_default_features_clist
6369 \NewDocumentCommand \defaultCJKfontfeatures { m }
6370 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
6371 \@onlypreamble \defaultCJKfontfeatures
6372 \NewDocumentCommand \addCJKfontfeatures { s O { } m }
6373 {
6374     \xeCJK_add_font_features:Nee #1 {#2} {#3}
6375     \tex_ignorespaces:D
6376 }
6377 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\xeCJK_add_font_features:Nnn 6378 \cs_new_protected:Npn \xeCJK_add_font_features:Nnn #1#2#3
6379 {
6380     \prop_get:NoNTF \g__xeCJK_family_font_name_prop
6381     \l_xeCJK_family_tl \l__xeCJK_font_name_tl
6382     {
6383         \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
6384         \seq_map_inline:Nn \g__xeCJK_sub_key_seq
6385         { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
6386         \seq_clear:N \l__xeCJK_sub_key_seq
6387         \clist_clear:N \l__xeCJK_add_block_features_clist
6388         \clist_map_function:nN {#2} \__xeCJK_add_sub_features:n
6389         \bool_lazy_and:nnT
6390         {#1}
6391         { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
6392         {
6393             \seq_map_function:NN
6394             \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
6395         }
6396     }
6397     \prop_get:NoNT \g__xeCJK_family_font_options_prop
6398     \l_xeCJK_family_tl \l__xeCJK_font_options_clist

```

```

6398     {
6399         \bool_lazy_or:nnT
6400         { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
6401         {#1}
6402     }
6403     \clist_concat:NNN \l__xeCJK_font_options_clist
6404     \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
6405 }
6406 \clist_concat:NNN \l__xeCJK_font_options_clist
6407 \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
6408 }
6409 \xeCJK_fontspec:oo \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
6410 }
6411 { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
6412 }
6413 \cs_new_protected:Npn \__xeCJK_add_sub_features:n #1
6414 {
6415     \seq_if_in:NnTF \g__xeCJK_sub_key_seq {#1}
6416     {
6417         \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
6418         \__xeCJK_add_sub_class_features:n {#1}
6419     }
6420     { \__xeCJK_warning:ne { SubBlock-undefined } {#1} }
6421 }
6422 \clist_new:N \l__xeCJK_add_font_features_clist
6423 \clist_new:N \l__xeCJK_add_block_features_clist
6424 \cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nee , Nne }
6425 \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
6426 {
6427     \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
6428     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
6429 }
\__xeCJK_add_sub_class_features:n 6430 \cs_new_protected:Npn \__xeCJK_add_sub_class_features:n #1
6431 {
6432     \prop_get:NoNTF \g__xeCJK_family_font_name_prop
6433     { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
6434     {
6435         \prop_get:NoN \g__xeCJK_family_font_options_prop
6436         { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
6437     }
6438     {
6439         \prop_get:NeNTF \g__xeCJK_family_font_name_prop
6440         { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
6441         {
6442             \prop_get:NeN \g__xeCJK_family_font_options_prop
6443             { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
6444         }
6445     }
6446     \prop_get:NoN \g__xeCJK_family_font_options_prop
6447     \l_xeCJK_family_tl \l__xeCJK_sub_font_options_clist
6448     \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl
6449 }
6450 }
6451 \clist_concat:NNN \l__xeCJK_sub_font_options_clist
6452 \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
6453 \clist_put_right:Ne \l__xeCJK_add_block_features_clist
6454 {
6455     #1 =
6456     {
6457         [ \exp_not:o \l__xeCJK_sub_font_options_clist ]
6458         { \exp_not:o \l__xeCJK_sub_font_name_tl }
6459     }
6460 }
6461 }
6462 \cs_generate_variant:Nn \prop_get:NnN { Ne }
6463 \prg_generate_conditional_variant:Nnn \prop_get:NnN { Ne } { T , TF }

```

```

LoadFandol 6464 \keys_define:nn { xeCJK / options }
6465 { LoadFandol .bool_gset:N = \g__xeCJK_fandol_bool }
6466 \cs_new_protected:Npn \__xeCJK_load_fandol:
6467 {
6468   \xeCJK_set_family:enn { \CJKrmdefault }
6469   { Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular }
6470   { FandolSong-Regular }
6471   \xeCJK_set_family:enn { \CJKsfdefault }
6472   { Extension = .otf , BoldFont = FandolHei-Bold } { FandolHei-Regular }
6473   \xeCJK_set_family:enn { \CJKttdefault }
6474   { Extension = .otf } { FandolFang-Regular }
6475 }

```

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。

```

6476 \__xeCJK_at_end_preamble:n
6477 {
6478   \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
6479   {
6480     \group_begin:
6481     \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
6482     \tl_gset:Ne \CJKfamilydefault
6483     {
6484       \str_case:onF { \familydefault }
6485       {
6486         { \rmdefault } { \exp_not:N \CJKrmdefault }
6487         { \sfdefault } { \exp_not:N \CJKsfdefault }
6488         { \ttdefault } { \exp_not:N \CJKttdefault }
6489       }
6490       { \CJKfamilydefault }
6491     }
6492     \group_end:
6493   }
6494   \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
6495   {
6496     \bool_if:NTF \g__xeCJK_fandol_bool
6497     {
6498       \__xeCJK_warning:n { fandol }
6499       \__xeCJK_load_fandol:
6500       \xeCJK_ensure_default_family:
6501     }
6502     { \__xeCJK_warning:ne { no-CJKfamily } { \CJKfamilydefault } }
6503   }
6504   { \xeCJK_ensure_default_family: }
6505 }
6506 \cs_new_protected:Npn \xeCJK_ensure_default_family:
6507 {
6508   \xeCJK_family_if_exist:eF { \CJKfamilydefault }
6509   {
6510     \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
6511     \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
6512     { \use:n }
6513     {
6514       \xeCJK_family_if_exist:eTF { \CJKrmdefault }
6515       { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
6516     }
6517   }
6518   \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
6519   {
6520     \prop_map_break:n
6521     { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
6522   }
6523 }

```

```

6524     \__xeCJK_warning:nne { CJKfamilydefault-undefined }
6525     { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
6526   }
6527   \xeCJK_switch_family:e { \CJKfamilydefault }
6528   \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
6529 }
6530 \__xeCJK_msg_new:nn { no-CJKfamily }
6531 {
6532   It~seems~that~you~have~not~declared~a~CJKfamily.\\
6533   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\
6534   \__xeCJK_msg_def_family_map:n {#1}'\\
6535   in~the~preamble~to~declare~the~default~CJKfamily.\\
6536 }
6537 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
6538 {
6539   Undefined~CJK~default~family~\__xeCJK_msg_family_map:n {#1}'~
6540   has~been~replaced~by~\__xeCJK_msg_family_map:n {#2}'\\.\\
6541   Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
6542 }
6543 \__xeCJK_msg_new:nn { fandol }
6544 {
6545   Fandol~is~being~set~as~the~default~font~for~CJK~text.\\
6546   Please~make~sure~it~has~been~properly~installed.
6547 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```
6548 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }
```

\setCJKmathfont 设置 CJK 数学字体。

```

6549 \NewDocumentCommand \setCJKmathfont { o m }
6550 {
6551   \__xeCJK_pass_args:nnnn
6552   { \xeCJK_set_family:nnn { \c__xeCJK_math_tl } } {#1} {#2}
6553   { }
6554 }
6555 \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。

```

6556 \cs_new_protected:Npn \xeCJK_set_mathfont:
6557 {
6558   \cs_if_exist_use:N \__xeCJK_save_um_char:
6559   \xeCJK_family_if_exist:eTF { \c__xeCJK_math_tl }
6560   { \__xeCJK_set_mathfont_aux: }
6561   {
6562     \xeCJK_family_if_exist:eT { \CJKfamilydefault }
6563     {
6564       \__xeCJK_copy_family:ee { \c__xeCJK_math_tl } { \CJKfamilydefault }
6565       \__xeCJK_set_mathfont_aux:
6566     }
6567   }
6568   \cs_if_exist_use:N \__xeCJK_restore_um_char:
6569 }
6570 \cs_new_protected:Npn \__xeCJK_set_mathfont_aux:
6571 {
6572   \tl_const:Nn \c__xeCJK_math_family_tl
6573   { \l__xeCJK_fontspec_family_tl }
6574   \xeCJK_declare_mathfont:ee
6575   { \c__xeCJK_math_tl }
6576   { \c__xeCJK_math_family_tl }
6577   \int_const:Nn \c__xeCJK_math_fam_int
6578   { \use:c { sym \c__xeCJK_math_tl } }
6579   \clist_gconcat:NNN \g__xeCJK_math_chars_clist

```

```

6580 \g__xeCJK_CJK_range_clist \g__xeCJK_FullLeft_range_clist
6581 \clist_gconcat:NNN \g__xeCJK_math_chars_clist
6582 \g__xeCJK_math_chars_clist \g__xeCJK_FullRight_range_clist
6583 \xeCJK_gset_mathcode:Nn \g__xeCJK_math_chars_clist
6584 { \c__xeCJK_math_fam_int }
6585 \xeCJK_set_mathfont_block:
6586 }
6587 \clist_new:N \g__xeCJK_math_chars_clist
6588 \prop_new:N \g__xeCJK_fam_prop

```

\xeCJK_set_mathfont_block: 分区数学字体。

```

6589 \cs_new_protected:Npn \xeCJK_set_mathfont_block:
6590 {
6591   \seq_if_empty:NF \g__xeCJK_CJK_sub_class_seq
6592   {
6593     \seq_map_function:NN
6594     \g__xeCJK_CJK_sub_class_seq
6595     \xeCJK_set_mathfont_block:n
6596   }
6597 }
6598 \cs_new_protected:Npn \xeCJK_set_mathfont_block:n #1
6599 {
6600   \xeCJK_block_family:nn { \c__xeCJK_math_tl } {#1}
6601   \prop_get:NoNTF \g__xeCJK_fam_prop
6602   \l__xeCJK_fontspec_family_tl \l__xeCJK_tmp_tl
6603   { \int_set:Nn \l__xeCJK_fam_int { \l__xeCJK_tmp_tl } }
6604   {
6605     \xeCJK_declare_mathfont:ee
6606     { \c__xeCJK_math_tl / #1 }
6607     { \l__xeCJK_fontspec_family_tl }
6608     \__xeCJK_set_mathfont_block_aux:cn
6609     { sym \c__xeCJK_math_tl / #1 } {#1}
6610   }
6611   \xeCJK_gset_mathcode:cn { g__xeCJK_CJK/#1_range_clist } { \l__xeCJK_fam_int }
6612 }
6613 \cs_new_protected:Npn \__xeCJK_set_mathfont_block_aux:Nn #1#2
6614 {
6615   \int_set_eq:NN \l__xeCJK_fam_int #1
6616   \prop_gput:Nnn \g__xeCJK_block_fam_prop {#2} {#1}
6617 }
6618 \int_new:N \l__xeCJK_fam_int
6619 \prop_new:N \g__xeCJK_block_fam_prop
6620 \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。

```

6621 \cs_new_protected:Npn \xeCJK_declare_mathfont:nn #1#2
6622 {
6623   \xeCJK_declare_symbol_font:nnnnn {#1} { \c__xeCJK_encoding_tl }
6624   {#2} { \mddefault } { \shapedefault }
6625   \cs_if_free:cF
6626   { \c__xeCJK_encoding_tl/#2/\bfdefault/\shapedefault }
6627   {
6628     \SetSymbolFont {#1} { bold } { \c__xeCJK_encoding_tl }
6629     {#2} { \bfdefault } { \shapedefault }
6630   }
6631   \prop_gput:Nne \g__xeCJK_fam_prop {#2} { \exp_not:c { sym #1 } }
6632 }
6633 \cs_generate_variant:Nn \prop_put:Nnn { Nne }
6634 \cs_generate_variant:Nn \prop_gput:Nnn { Nne }
6635 \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。

```

6636 \cs_new_protected:Npn \xeCJK_declare_symbol_font:nnnnn #1

```

```

6637 { \__xeCJK_declare_symbol_font:cnnnn { sym #1 } }
6638 \cs_new_protected:Npn \__xeCJK_declare_symbol_font:Nnnnn #1
6639 {
6640   \xeCJK_new_fam:N #1
6641   \xeCJK_new_symbol_font:Nnnnn #1
6642 }
6643 \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 $\text{\LaTeX} 2_{\epsilon}$ 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。

```

6644 \cs_new_protected:Npn \xeCJK_new_fam:N #1
6645 {
6646   \int_compare:nNtF
6647     { \g__xeCJK_fam_allocation_int } > { \g__xeCJK_fam_bottom_int }
6648     {
6649       \int_set_eq:NN \allocationnumber \g__xeCJK_fam_allocation_int
6650       \int_const:Nn #1 { \allocationnumber }
6651       \iow_log:e
6652       {
6653         \token_to_str:N #1 =
6654         \token_to_str:N \mathgroup \int_use:N \allocationnumber
6655       }
6656       \int_gdecr:N \g__xeCJK_fam_allocation_int
6657     }
6658     { \__xeCJK_error:n { fam-exhausted } }
6659 }
6660 \tex_countdef:D \g__xeCJK_fam_bottom_int = 18 ~
6661 \int_new:N \g__xeCJK_fam_allocation_int
6662 \int_gset:Nn \g__xeCJK_fam_allocation_int { 255 }
6663 \__xeCJK_msg_new:nn { fam-exhausted }
6664 { 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
6665 \cs_new_protected:Npn \xeCJK_new_symbol_font:Nnnnn #1#2#3#4#5
6666 { \__xeCJK_new_symbol_font:Nc #1 { #2/#3/#4/#5 } }
6667 \cs_new_protected:Npn \__xeCJK_new_symbol_font:NN #1#2
6668 {
6669   \tl_put_right:Nn \group@list { \group@elt #1 #2 }
6670   \cs_set:Npn \version@elt ##1
6671     { \tl_put_right:Nn ##1 { \getanddefine@fonts #1 #2 } }
6672   \version@list
6673 }
6674 \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
6675 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nn #1#2
6676 {
6677   \clist_map_inline:Nn #1
6678     {
6679       \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn { ##1 }
6680       { 0 } {#2}
6681     }
6682 }
6683 \cs_generate_variant:Nn \xeCJK_gset_mathcode:Nn { c }
6684 \cs_new_protected:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
6685 {
6686   \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
6687   \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
6688   {
6689     \xeCJK_gset_mathcode:Nnn \l__xeCJK_begin_int {#3} {#4}
6690     \int_incr:N \l__xeCJK_begin_int
6691   }
6692 }
6693 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nnn #1#2#3
6694 { \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 L^AT_EX 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 L^AT_EX 的环境定义里使用 \begingroup 和 \endgroup 来分组。

```

6695 \int_new:N \l__xeCJK_verb_case_int
6696 \keys_define:nn { xeCJK / options }
6697 {
6698   Verb .choices:nn =
6699     { true , env+ , env , false }
6700     { \int_set_eq:NN \l__xeCJK_verb_case_int \l_keys_choice_int } ,
6701   Verb .default:n = { env }
6702 }
6703 \cs_new_protected:Npn \__xeCJK_verb_font_hook:
6704 {
6705   \if_case:w \l__xeCJK_verb_case_int
6706   \or:
6707     \__xeCJK_nobreak_skip_zero:
6708   \or:
6709     \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
6710       { \xeCJKVerbAddon }
6711       { \__xeCJK_nobreak_skip: }
6712   \or:
6713     \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
6714       { \xeCJKVerbAddon }
6715       { \__xeCJK_nobreak_skip_zero: }
6716   \fi:
6717 }
6718 \__xeCJK_after_preamble:n
6719 {
6720   \cs_set_protected:Npe \verbatim@font
6721     { \exp_not:o { \verbatim@font } \__xeCJK_verb_font_hook: }
6722 }
\__xeCJK_nobreak_skip_zero: 6723 \cs_new_protected:Npn \__xeCJK_nobreak_skip_zero:
\__xeCJK_nobreak_skip: 6724 {
6725   \__xeCJK_reset_shipout_skip:
6726   \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
6727   \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
6728   \tl_put_right:Nn \l__xeCJK_reset_shipout_skip_hook_tl
6729     {
6730       \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
6731       \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
6732     }
6733   \xeCJK_cs_clear:N \CJKglue
6734   \xeCJK_cs_clear:N \CJKecglue
6735   \xeCJK_cs_clear:N \xeCJK_check_for_glue:
6736   \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \xeCJK_class_group_end:
6737   \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
6738   \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
6739 }
6740 \cs_new_protected:Npn \__xeCJK_nobreak_skip:
6741 {
6742   \__xeCJK_reset_shipout_skip:
6743   \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_ccglue_skip
6744   \skip_if_eq:nnTF { \l__xeCJK_ccglue_skip } { \c_zero_skip }
6745     { \xeCJK_cs_clear:N \CJKglue }
6746     { \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue: }
6747   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
6748   \skip_if_eq:nnTF { \l__xeCJK_ecglue_skip } { \c_zero_skip }
6749     { \xeCJK_cs_clear:N \CJKecglue }
6750     { \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue: }
6751   \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
6752   \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
6753 }
6754 \cs_new_protected:Npn \__xeCJK_nobreak_ccglue:
6755 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ccglue_skip }

```

```

6756 \cs_new_protected:Npn \__xeCJK_nobreak_ecglue:
6757 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ecglue_skip }

\__xeCJK_reset_shipout_skip: 6758 \cs_new_protected:Npn \__xeCJK_reset_shipout_skip:
6759 {
6760   \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
6761   \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
6762   \cs_set_eq:NN \__xeCJK_shipout_punct_hskip:n \__xeCJK_punct_hskip:n
6763   \cs_set_eq:NN
6764     \__xeCJK_shipout_punct_breakable_kern:n \__xeCJK_punct_breakable_kern:n
6765   \tl_set:Nx \l__xeCJK_off_verb_addon_tl
6766   {
6767     \bool_if:NTF \l__xeCJK_xecglue_bool
6768       { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
6769     { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
6770     \exp_not:n
6771     {
6772       \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
6773       \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
6774       \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_shipout_punct_hskip:n
6775       \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n
6776         \__xeCJK_shipout_punct_breakable_kern:n
6777       \l__xeCJK_reset_shipout_skip_hook_tl
6778     }
6779   }
6780   \xeCJK_add_to_shipout:n { \l__xeCJK_off_verb_addon_tl }
6781   \keys_set:nn { xeCJK / options } { xCJKecglue = false }
6782 }
6783 \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。

```

6784 \NewDocumentCommand \xeCJKVerbAddOn { }
6785 {
6786   \int_compare:nNnF \tex_currentgrouplevel:D = \c_zero_int
6787   {
6788     \bool_if:NF \l__xeCJK_listings_env_bool
6789     {
6790       \dim_compare:nNnTF
6791       { \tex_fontdimen:D 2 ~ \tex_font:D } =
6792       { \tex_fontcharwd:D \tex_font:D \c__xeCJK_mono_letter_int }
6793       {
6794         \__xeCJK_set_verb_exspace:
6795         \__xeCJK_verb_addon:
6796       }
6797       {
6798         \int_if_odd:nTF { \l__xeCJK_verb_case_int }
6799         { \__xeCJK_nobreak_skip_zero: }
6800         { \__xeCJK_nobreak_skip: }
6801       }
6802     }
6803   }
6804 }
6805 \int_const:Nn \c__xeCJK_mono_letter_int { 77 }
6806 \bool_new:N \l__xeCJK_listings_env_bool
6807 \NewDocumentCommand \xeCJKOffVerbAddOn { }
6808 { \tl_use:N \l__xeCJK_off_verb_addon_tl }
6809 \tl_new:N \l__xeCJK_off_verb_addon_tl
6810 \cs_new_protected:Npn \__xeCJK_verb_addon:
6811 {
6812   \bool_if:NF \l__xeCJK_verb_addon_bool
6813   { \__xeCJK_verb_addon_action: }
6814   \skip_if_eq:nnTF { \l__xeCJK_verb_exspace_skip } { \c_zero_skip }
6815   {

```

```

6816         \xeCJK_cs_clear:N \CJKglue
6817         \xeCJK_cs_clear:N \CJKecglue
6818     }
6819     {
6820         \skip_set_eq:NN \l__xeCJK_ccglue_skip \l__xeCJK_verb_exspace_skip
6821         \skip_set:Nn \l__xeCJK_ecglue_skip { \l__xeCJK_verb_exspace_skip / 2 }
6822         \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue:
6823         \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue:
6824     }
6825     \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
6826     \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_CJK_and_Boundary:w
6827 }
6828 \cs_new_protected:Npn \__xeCJK_verb_addon_action:
6829 {
6830     \bool_set_true:N \l__xeCJK_verb_addon_bool
6831     \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
6832     \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
6833     \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
6834     \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
6835     \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
6836     \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
6837     \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
6838     \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
6839     \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
6840     \cs_set_protected:Npe \xeCJKOffVerbAddon
6841     {
6842         \__xeCJK_reset_char_class:n { FullLeft }
6843         \__xeCJK_reset_char_class:n { FullRight }
6844         \__xeCJK_reset_char_class:n { HalfLeft }
6845         \__xeCJK_reset_char_class:n { HalfLeft }
6846         \__xeCJK_reset_char_class:n { NormalSpace }
6847         \bool_if:NTF \l__xeCJK_xecglue_bool
6848         { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
6849         { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
6850         \exp_not:n
6851         {
6852             \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
6853             \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
6854             \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
6855             \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
6856         }
6857     }
6858     \xeCJK_add_to_shipout:n { \xeCJKOffVerbAddon }
6859     \keys_set:nn { xeCJK / options } { xCJKecglue = false }
6860 }
6861 \cs_new_protected:Npn \__xeCJK_verb_CJK_and_Boundary:w
6862 { \xeCJK_class_group_end: \CJKecglue }
6863 \cs_new_protected:Npn \__xeCJK_reset_char_class:n #1
6864 {
6865     \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
6866     \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
6867     { \tex_XeTeXcharclass:D ##1 = \l__xeCJK_tmp_int }
6868 }
6869 \bool_new:N \l__xeCJK_verb_addon_bool
6870 \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。

```

6871 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:
6872 {
6873     \tl_if_exist:cTF { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
6874     {
6875         \skip_set:Nn \l__xeCJK_verb_exspace_skip
6876         { \use:c { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size } }
6877     }
6878     {

```

```

6879 \tl_set:Nn \l__xeCJK_current_coor_tl { \CJK@family/\curr@fontshape }
6880 \prop_get:NnTF \g__xeCJK_scale_family_prop
6881 \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
6882 {
6883   \xeCJK_switch_family:o { \l_xeCJK_family_tl }
6884   \skip_zero:N \l__xeCJK_verb_exspace_skip
6885 }
6886 {
6887   \group_begin: \xeCJK_select_font: \exp_args:NNo \group_end:
6888   \__xeCJK_set_verb_exspace:n
6889   { \dim_use:N \tex_fontcharwd:D \tex_font:D "4E00 ~ }
6890 }
6891 }
6892 }
6893 \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。

```

6894 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:n #1
6895 {
6896   \skip_set:Nn \l__xeCJK_verb_exspace_skip
6897   { 2 \tex_fontdimen:D 2 ~ \tex_font:D - #1 }
6898   \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
6899   {
6900     \skip_zero:N \l__xeCJK_verb_exspace_skip
6901     \exp_args:Nee \__xeCJK_set_verb_scale:nn
6902     { \dim_to_fp:n { 2 \tex_fontdimen:D 2 ~ \tex_font:D } }
6903     { \dim_to_fp:n {#1} }
6904   }
6905   {
6906     \tl_const:ce { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
6907     { \skip_use:N \l__xeCJK_verb_exspace_skip }
6908   }
6909 }

```

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

```

6910 \cs_new_protected:Npn \__xeCJK_set_verb_scale:nn #1#2
6911 {
6912   \fp_set:Nn \l__xeCJK_scale_factor_fp { #1 / #2 }
6913   \__xeCJK_warning:nee { scale-factor }
6914   { \fp_eval:n { trunc ( \l__xeCJK_scale_factor_fp , 4 ) } }
6915   { \fp_eval:n { ceil ( #2 / #1 , 4 ) } }
6916   \xeCJK_add_font_features:Nne \c_true_bool
6917   { } { Scale = { \fp_use:N \l__xeCJK_scale_factor_fp } }
6918   \prop_gput:Noo \g__xeCJK_scale_family_prop
6919   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
6920 }
6921 \__xeCJK_msg_new:nn { scale-factor }
6922 {
6923   \token_to_str:N \xeCJKVerbAddon'~may~not~work~properly.\\
6924   You~may~set~`Scale=#1'~to~CJKfamily~
6925   \__xeCJK_msg_family_map:n { \l_xeCJK_family_tl },\\
6926   or~set~`Scale=#2'~to~family~
6927   \str_if_eq:eeTF \f@family \ttdefault
6928   { \token_to_str:N \ttdefault } { \f@family }'.
6929 }
6930 \fp_new:N \l__xeCJK_scale_factor_fp
6931 \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, 这时可视空格按照 OT1 编码传统一般就是字体中的\char32。

```

6932 \cs_new_protected:Npn \xeCJK_setup_visible_space:
6933 {
6934   \xeCJK_make_boundary:
6935   \xeCJK_glyph_if_exist:Ntf { ^^^2423 }
6936   { \tl_set:Nn \l__xeCJK_visible_space_tl { ^^^2423 } }

```

```

6937 {
6938   \int_compare:nNnTF { \tex_XeTeXfonttype:D \tex_font:D } = \c_zero_int
6939   {
6940     \tl_set:Nc \l__xeCJK_visible_space_tl
6941       {
6942         \str_if_eq:eeTF { \f@family } { \ttdefault }
6943         { \c_catcode_other_space_tl }
6944         { \exp_not:N \textvisiblespace }
6945       }
6946     }
6947     { \__xeCJK_visible_space_fallback: }
6948   }
6949   \cs_set_eq:NN \xobeysp \l__xeCJK_visible_space_tl
6950 }
6951 \tl_new:N \l__xeCJK_visible_space_tl
6952 \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。

```

6953 \cs_new_protected:Npn \__xeCJK_visible_space_fallback:
6954 {
6955   \exp_args:Nc \__xeCJK_visible_space_fallback_auxi:N
6956     { xeCJK/space/\curr@fontshape/\f@size }
6957 }
6958 \cs_new_protected:Npn \__xeCJK_visible_space_fallback_auxi:N #1
6959 {
6960   \cs_if_exist:NF #1
6961   { \__xeCJK_visible_space_fallback_auxii:N #1 }
6962   \tl_set:Nn \l__xeCJK_visible_space_tl {#1}
6963 }

```

`\__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。

```

6964 \cs_new_protected:Npn \__xeCJK_visible_space_fallback_auxii:N #1
6965 {
6966   \group_begin:
6967   \exp_args:No \__xeCJK_set_visible_space_size:n
6968     { \dim_use:N \tex_fontdimen:D 2 ~ \tex_font:D }
6969   \cs_new_protected:Npe #1
6970     { \group_begin: \tex_the:D \tex_font:D ~~~~2423 \group_end: }
6971   \group_end:
6972 }
6973 \cs_new_protected:Npn \__xeCJK_set_visible_space_size:n #1
6974 {
6975   \fontencoding { \UnicodeEncodingName }
6976   \tl_set:Nn \f@family { lmtt }
6977   \selectfont
6978   \dim_compare:nNnF {#1} = { \tex_fontdimen:D 2 ~ \tex_font:D }
6979   {
6980     \fontsize
6981     {
6982       \dim_eval:n
6983       {
6984         \f@size pt *
6985         \dim_ratio:nn {#1} { \tex_fontdimen:D 2 ~ \tex_font:D }
6986       }
6987     }
6988     { \f@baselineskip }
6989   \selectfont
6990 }
6991 }

```

5.16 xeCJK 其他选项

LocalConfig 声明载入本地配置文件的选项。

```

6992 \keys_define:nn { xeCJK / options }
6993 {
6994   LocalConfig .choice: ,
6995   LocalConfig / false .code:n =
6996     { \bool_gset_false:N \g__xeCJK_config_bool } ,
6997   LocalConfig / true .code:n =
6998     {
6999       \bool_gset_true:N \g__xeCJK_config_bool
7000       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
7001     } ,
7002   LocalConfig / unknown .code:n =
7003     {
7004       \bool_gset_true:N \g__xeCJK_config_bool
7005       \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
7006     } ,
7007   LocalConfig .default:n = { true }
7008 }
7009 \tl_new:N \g__xeCJK_config_name_tl
7010 \bool_new:N \g__xeCJK_config_bool

```

CJKnumber 和 **indentfirst** 是过时选项。

```

7011 \keys_define:nn { xeCJK / options }
7012 {
7013   CJKnumber .code:n =
7014     { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { CJKnumb } } ,
7015   indentfirst .code:n =
7016     { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { indentfirst } } ,
7017   normalindentfirst .code:n =
7018     { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { } }
7019 }
7020 \__xeCJK_msg_new:nn { option-deprecated }
7021 {
7022   The~`#1'~option~is~deprecated.\\
7023   \tl_if_empty:nF {#2}
7024     { You~may~load~the~package~`#2'~after~xeCJK~to~use~its~function.\\ }
7025 }

```

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。

```

7026 \keys_define:nn { xeCJK / options }
7027 {
7028   quiet .code:n =
7029     {
7030       \msg_redirect_module:nnn { xeCJK } { warning } { info }
7031       \msg_redirect_module:nnn { xeCJK } { info } { none }
7032       \xeCJK_if_package_loaded:nF { fontspec }
7033         { \PassOptionsToPackage { quiet } { fontspec } }
7034     } ,
7035   silent .code:n =
7036     {
7037       \msg_redirect_module:nnn { xeCJK } { warning } { none }
7038       \msg_redirect_module:nnn { xeCJK } { info } { none }
7039       \xeCJK_if_package_loaded:nF { fontspec }
7040         { \PassOptionsToPackage { silent } { fontspec } }
7041     } ,
7042   unknown .code:n =
7043     {
7044       \xeCJK_if_package_loaded:nTF { fontspec }
7045         { \__xeCJK_error:ne { key-unknown } { \l_keys_key_str } }
7046         { \PassOptionsToPackage { \l_keys_key_str } { fontspec } }
7047     }
7048 }

```

```

7049 \__xeCJK_msg_new:nn { key-unknown }
7050 {
7051   Sorry,~but~xeCJK/options~does~not~have~a~key~called~`#1'.\\
7052   The~key~`#1'~is~being~ignored.
7053 }

```

5.17 xeCJK 初始化设置

```

\CJKsymbol 7054 \cs_new_eq:NN \CJKsymbol \use:n
\CJKpunctsymbol 7055 \cs_new_eq:NN \CJKpunctsymbol \use:n

```

xeCJK 宏包的初始化设置。

```

7056 \keys_set:nn { xeCJK / options }
7057 {
7058   CJKglue = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
7059   CJKecglue = { ~ } ,
7060   xCJKecglue = false ,
7061   CheckSingle = false ,
7062   PlainEquation = false ,
7063   CheckFullRight = false ,
7064   CJKspace = false ,
7065   CJKmath = false ,
7066   xeCJKactive = true ,
7067   LocalConfig = true ,
7068   LoadFandol = true ,
7069   RubberPunctSkip = true ,
7070   Verb = env ,
7071   EmboldenFactor = 4 ,
7072   SlantFactor = 0.167 ,
7073   PunctStyle = quanjiao ,
7074   NewLineCS = { \par \[ } ,
7075   EnvCS = { \begin \end } ,
7076   WidowPenalty = { 10 000 } ,
7077   NoBreakCS = { \footnote \footnotemark \nobreak } ,
7078   KaiMingPunct = { ~~~~~3002 ~~~~~ff0e ~~~~~ff1f ~~~~~ff01 } ,
7079   LongPunct = { ~~~~~2014 ~~~~~2e3a ~~~~~2025 ~~~~~2026 } ,
7080   NoBreakLongPunct = { ~~~~~2025 ~~~~~2026 } ,
7081   MiddlePunct = { ~~~~~2013 ~~~~~2014 ~~~~~2e3a ~~~~~2027 ~~~~~00b7 ~~~~~30fb ~~~~~ff65 } ,
7082   AllowBreakBetweenPuncts = false
7083 }
7084 \defaultCJKfontfeatures { Script = CJK }

```

半字线连接号¹⁶应为半角宽度。

```

7085 \xeCJKsetwidth { ~~~~~2013 } { 0.5 em }

```

执行宏包选项,并载入 fontspec 宏包。

```

7086 \cs_if_exist:NTF \ProcessKeyOptions
7087 { \ProcessKeyOptions [ xeCJK / options ] }
7088 {
7089   \RequirePackage { l3keys2e }
7090   \ProcessKeysOptions { xeCJK / options }
7091 }
7092 \RequirePackage { fontspec } [ 2020/02/03 ]

```

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

```

7093 \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。

```

7094 \keys_define:nn { xeCJK / options }
7095 {
7096   LocalConfig .code:n =
7097     { \__xeCJK_warning:ne { option-invalid } { \l_keys_key_str } }
7098 }

```

¹⁶见《夹用英文的中文文本的标点符号用法(草案)》5.13 节。

```

7099 \__xeCJK_msg_new:nn { option-invalid }
7100 {
7101   The~`#1'~option~can~only~be~set~in~the~optional~argument~to~the\\
7102   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
7103   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
7104 }

\CJKrmdefault 7105 \tl_if_exist:NF \CJKrmdefault
\JKsfdefault 7106 { \tl_new:N \CJKrmdefault \tl_gset:Nn \CJKrmdefault { rm } }
\JKttdefault 7107 \tl_if_exist:NF \JKsfdefault
\JKfamilydefault 7108 { \tl_new:N \JKsfdefault \tl_gset:Nn \JKsfdefault { sf } }
7109 \tl_if_exist:NF \JKttdefault
7110 { \tl_new:N \JKttdefault \tl_gset:Nn \JKttdefault { tt } }
7111 \tl_new:N \l__xeCJK_family_default_init_tl
7112 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
7113 \tl_set:Nx \l__xeCJK_family_default_init_tl
7114 {
7115   \exp_not:N \__xeCJK_family_default_wrap:n
7116   {
7117     \tl_if_exist:NTF \JKfamilydefault
7118     { \exp_not:o \JKfamilydefault }
7119     { \exp_not:N \CJKrmdefault }
7120   }
7121 }
7122 \tl_if_exist:NF \JKfamilydefault
7123 { \tl_new:N \JKfamilydefault }
7124 \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。

```

7125 \NewDocumentCommand \xeCJKsetup { +m }
7126 {
7127   \keys_set:nn { xeCJK / options } {#1}
7128   \tex_ignorespaces:D
7129 }

\xeCJKsetemboldenfactor 7130 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
\xeCJKsetslantfactor 7131 { \xeCJKsetup { EmboldenFactor = {#1} } }
7132 \NewDocumentCommand \xeCJKsetslantfactor { m }
7133 { \xeCJKsetup { SlantFactor = {#1} } }

\punctstyle 7134 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
\xeCJKplainchr 7135 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }

\JKsetecglue 7136 \NewDocumentCommand \JKsetecglue { m } { \xeCJKsetup { JKecglue = {#1} } }
7137 \cs_new_eq:NN \xeCJKsetecglue \JKsetecglue

\JKspace 7138 \NewDocumentCommand \JKspace { } { \xeCJKsetup { JKspace = true } }
\JKnospace 7139 \NewDocumentCommand \JKnospace { } { \xeCJKsetup { JKspace = false } }

\xeCJKallowbreakbetweenpuncts 7140 \NewDocumentCommand \xeCJKallowbreakbetweenpuncts { }
\xeCJKnobreakbetweenpuncts 7141 { \xeCJKsetup { AllowBreakBetweenPuncts = true } }
7142 \NewDocumentCommand \xeCJKnobreakbetweenpuncts { }
7143 { \xeCJKsetup { AllowBreakBetweenPuncts = false } }

\xeCJKenablefallback 7144 \NewDocumentCommand \xeCJKenablefallback { }
\xeCJKdisablefallback 7145 { \xeCJKsetup { AutoFallBack = true } }
7146 \NewDocumentCommand \xeCJKdisablefallback { }
7147 { \xeCJKsetup { AutoFallBack = false } }

\xeCJKsetcharclass 7148 \__xeCJK_msg_new:nn { xeCJKsetcharclass-deprecated }
7149 {
7150   \token_to_str:N \xeCJKsetcharclass\ is~deprecated~and~
7151   does~not~update~CJKmath~char~lists.\\
7152   Use~\token_to_str:N \xeCJKDeclareCharClass\ instead.
7153 }
7154 \NewDocumentCommand \xeCJKsetcharclass { m m m }
7155 {
7156   \__xeCJK_error:n { xeCJKsetcharclass-deprecated }
7157   \xeCJK_set_char_class:nnn {#1} {#2} {#3}
7158   \xeCJKResetPunctClass
7159 }

```

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。

```

7160 \cs_new_protected:Npn \xeCJK@update@fam
7161 {
7162   \addto@hook \everymath
7163   {
7164     \__xeCJK_update_main_fam:
7165     \__xeCJK_update_block_fam:
7166   }
7167 }
7168 \cs_new_protected:Npn \__xeCJK_update_main_fam:
7169 {
7170   \group_begin:
7171   \xeCJK_select_font:
7172   \exp_last_unbraced:NNNo \group_end:
7173   \tex_textfont:D \c_xeCJK_math_fam_int \tex_the:D \tex_font:D
7174 }
7175 \cs_new_protected:Npn \__xeCJK_update_block_fam:
7176 {
7177   \prop_if_empty:NF \g__xeCJK_block_fam_prop
7178   {
7179     \prop_map_function:NN
7180     \g__xeCJK_block_fam_prop
7181     \__xeCJK_update_block_fam:nn
7182   }
7183 }
7184 \cs_new_protected:Npn \__xeCJK_update_block_fam:nn #1#2
7185 {
7186   \int_set:Nn \l__xeCJK_fam_int {#2}
7187   \group_begin:
7188   \xeCJK_select_font:n {#1}
7189   \exp_last_unbraced:NNNo \group_end:
7190   \tex_textfont:D \l__xeCJK_fam_int \tex_the:D \tex_font:D
7191 }
7192 \__xeCJK_after_end_preamble:n
7193 {
7194   \bool_lazy_and:nnT
7195   { \g__xeCJK_math_bool }
7196   { \cs_if_exist_p:N \Url@MathSetup }
7197   { \tl_put_right:Nn \Url@MathSetup { \xeCJK@update@fam } }
7198 }

```

`\(` 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。

```

7199 \cs_new_protected:Npn \__xeCJK_math_robust:N #1
7200 {
7201   \group_begin: \exp_args:NcNc \group_end:
7202   { \__xeCJK_math_robust_aux:NN } #1 { \cs_to_str:N #1 ~ }

```

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

```

7203 }
7204 \cs_new_protected:Npn \__xeCJK_math_robust_aux:NN #1#2
7205 {
7206   \exp_args:Ne \str_case:nnTF { \cs_replacement_spec:N #1 }
7207   {
7208     { \x@protect #1 \protect #2 } { }
7209     { \protect #2 } { }
7210   }
7211   { \__xeCJK_math_robust:NN #1#2 }
7212   { \__xeCJK_math_robust:NN #1#1 }
7213 }
7214 \cs_new_protected:Npn \__xeCJK_math_robust:NN #1#2
7215 {
7216   \str_if_eq:eeTF { \cs_argument_spec:N #2 } { }
7217   {
7218     \exp_args:No \tl_if_head_eq_meaning:nNTF {#2} \scan_stop:
7219     {
7220       \cs_gset_protected:Npe #1
7221       { \tl_tail:N #2 }
7222     }
7223     {
7224       \cs_if_eq:NNTF #1 \ensuremath
7225       {
7226         \cs_gset_protected:Npe #1
7227         { \exp_not:o {#2} }
7228       }
7229       {
7230         \__xeCJK_warning:nee { robust-failure }
7231         { \token_to_str:N #1 } { \token_to_meaning:N #2 }
7232       }
7233     }
7234   }
7235   {
7236     \__xeCJK_warning:nee { robust-failure }
7237     { \token_to_str:N #1 } { \token_to_meaning:N #2 }
7238   }
7239 }
7240 \__xeCJK_msg_new:nnn { robust-failure }
7241 { xeCJK~can~not~make~`#1'~robust. }
7242 {
7243   The~current~meaning~of~`#1'~is:\\
7244   \iow_indent:n {#2}
7245 }
7246 \cs_if_eq:NNTF \(\ \math
7247 {
7248   \__xeCJK_math_robust:N \(\
7249   \cs_set_eq:NN \math \(\
7250 }
7251 {
7252   \__xeCJK_math_robust:N \(\
7253   \__xeCJK_math_robust:N \math
7254 }
7255 \cs_if_eq:NNTF \) \endmath
7256 {
7257   \__xeCJK_math_robust:N \)
7258   \cs_set_eq:NN \endmath \)
7259 }
7260 {
7261   \__xeCJK_math_robust:N \)
7262   \__xeCJK_math_robust:N \endmath
7263 }
7264 \__xeCJK_math_robust:N \ensuremath

```

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;它们不直接向所有外层 capture 报告,否则原语 \setbox 中不可见的测量公式也会被当成外层输出。已经注册的命令分别检查可见正文的开头和结尾:

开头是 $\$, \backslash($ 或 $\backslash\text{ensuremath}$ 时, 在 math-on 前报告首类别; 结尾是对应公式时, 把 math 记为当前 capture 的确定末类别。 $\backslash\text{ensuremath}$ 已在数学模式中时不重复处理当前列表。

```

7265 \clist_map_inline:nn { ( , math , ensuremath }
7266 { \__xeCJK_boundary_reserve:n {#1} }
7267 \cs_if_eq:NNTF \ ( \math
7268 {
7269   \cs_new_eq:NN \__xeCJK_orig_inline_math_begin: \ (
7270   \cs_gset_protected:Npn \ (
7271     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_inline_math_begin: }
7272   \cs_gset_eq:NN \math \ (
7273 }
7274 {
7275   \cs_new_eq:NN \__xeCJK_orig_inline_math_begin: \ (
7276   \cs_gset_protected:Npn \ (
7277     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_inline_math_begin: }
7278   \cs_new_eq:NN \__xeCJK_orig_math_begin: \math
7279   \cs_gset_protected:Npn \math
7280     { \__xeCJK_boundary_math_begin: \__xeCJK_orig_math_begin: }
7281 }
7282 \cs_new_eq:NN \__xeCJK_orig_ensuremath: \ensuremath
7283 \cs_gset_protected:Npn \ensuremath
7284 {
7285   \mode_if_math:TF
7286   { \__xeCJK_orig_ensuremath: }
7287   { \__xeCJK_boundary_math_begin: \__xeCJK_orig_ensuremath: }
7288 }

```

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, 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, 结尾公式在正文参数实际排完时确认。普通正文仍由 interchar 转换报告实际首尾类别。

```

7289 \cs_new_protected:Npn \__xeCJK_boundary_register_text_command:N #1
7290 {
7291   \group_begin: \exp_args:NcNc \group_end:
7292   { \__xeCJK_boundary_register_text_command:NN }
7293   #1 { \cs_to_str:N #1 ~ }
7294 }
7295 \cs_new_protected:Npn \__xeCJK_boundary_register_text_command:NN #1#2
7296 {
7297   \__xeCJK_boundary_reserve:N #1
7298   \__xeCJK_boundary_reserve:N #2
7299   \exp_args:Nc \cs_new_eq:NN
7300   { \__xeCJK_boundary_orig:\cs_to_str:N #1 :n } #2
7301   \cs_gset_protected:Npn #2 ##1
7302   {
7303     \__xeCJK_boundary_if_math_edge:nTF {##1}
7304     {
7305       \__xeCJK_boundary_inline_stream_begin:n { auto }
7306       \__xeCJK_boundary_math_begin:n {##1}
7307       \use:c { \__xeCJK_boundary_orig:\cs_to_str:N #1 :n }
7308       { ##1 \__xeCJK_boundary_math_end:n {##1} }
7309       \__xeCJK_boundary_inline_stream_end:n { auto }
7310     }
7311     { \use:c { \__xeCJK_boundary_orig:\cs_to_str:N #1 :n } {##1} }
7312   }
7313 }
7314 \clist_map_inline:nn
7315 {
7316   \textrm , \textsf , \texttt , \textnormal , \textbf , \textmd ,
7317   \textit , \textsl , \textsc , \textup , \emph
7318 }
7319 { \__xeCJK_boundary_register_text_command:N #1 }

```

$\backslash\text{fontfamily}$ 使用 LaTeX_{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。
 $\backslash\text{xeCJK@family}$

```

7320 \cs_set_eq:NN \xeCJK@family \xeCJK_switch_family:e

```

```

7321 \ctex_gadd_ltxhook:nn { rmfamily } { \xeCJK@family { \CJKrmdefault } }
7322 \ctex_gadd_ltxhook:nn { sffamily } { \xeCJK@family { \CJKsfdefault } }
7323 \ctex_gadd_ltxhook:nn { ttfamily } { \xeCJK@family { \CJKttdefault } }
7324 \ctex_gadd_ltxhook:nn { normalfont } { \xeCJK@family { \CJKfamilydefault } }

7325 <@@=>

```

`\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 `xCJKecglue` 选项,第一个空格也可以被省略。事实上,在 `\sw@slant` 的定义中, `@@@italiccorr` 前面的 `\lastskip` 和 `\lastpenalty` 有四种情况,这里只对它们都为零的情况进行处理。

```

7326 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
7327 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
7328 \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。

```

7329 \cs_new_protected:Npn \xeCJK@italiccorr
7330 {
7331   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7332   { \xeCJK_italic_correction: }
7333   { \@@italiccorr }
7334 }

7335 <@@=\xeCJK>

```

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

```

7336 \cs_new_protected:Npn \xeCJK_italic_correction:
7337 { \__xeCJK_if_last_kern:T { \__xeCJK_italic_correction: } }
7338 \cs_new_protected:Npn \__xeCJK_italic_correction:
7339 {
7340   \dim_case:nnF { \tex_lastkern:D }
7341   {
7342     { \__xeCJK_node:n { default } }
7343     {
7344       \xeCJK_remove_node: \tex_italiccorrection:D
7345       \xeCJK_make_node:n { default }
7346     }
7347     { \__xeCJK_node:n { CJK } }
7348     {
7349       \xeCJK_remove_node: \tex_italiccorrection:D
7350       \xeCJK_make_node:n { CJK }
7351       \__xeCJK_italic_correction_aux:
7352     }
7353     { \__xeCJK_node:n { CJK-space } }
7354     {
7355       \xeCJK_remove_node: \tex_italiccorrection:D
7356       \xeCJK_make_node:n { CJK-space }
7357       \__xeCJK_italic_correction_aux:
7358     }
7359   }
7360   { \tex_italiccorrection:D }
7361 }

```

`\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:`)。

```

7362 \cs_new_protected:Npn \__xeCJK_italic_correction_aux:
7363 {

```

```

7364             \exp_after:wN \exp_after:wN \exp_after:wN
7365     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7366     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7367     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
7368     \xeCJK_ignore_spaces:w
7369 }

```

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

```
7370 \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。

```

7371 \__xeCJK_after_end_preamble:n
7372 {
7373     \int_compare:nNnF
7374     { \c_xeCJK_class_begin_int + \seq_count:N \g_xeCJK_new_class_seq } =
7375     { \g_xeCJK_xetex_allocator_int }
7376     {
7377         \int_step_inline:nnn
7378         { \c_xeCJK_class_begin_int + 1 }
7379         { \g_xeCJK_xetex_allocator_int }
7380         {
7381             \seq_if_in:NnF \g_xeCJK_new_class_seq {#1}
7382             { \__xeCJK_set_others_toks:n {#1} }
7383         }
7384     }
7385 }
7386 \cs_new_protected:Npn \__xeCJK_set_others_toks:n #1
7387 {
7388     \int_set:cn { \__xeCJK_class_csname:n { Others } } {#1}
7389     \seq_map_inline:Nn \g_xeCJK_CJK_class_seq
7390     {
7391         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
7392         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
7393         \xeCJK_app_inter_class_toks:nne {##1} { Others }
7394         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
7395         \xeCJK_pre_inter_class_toks:nne { Others } {##1}
7396         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
7397         \tl_if_blank:eT
7398         { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
7399         {
7400             \xeCJK_copy_inter_class_toks:nnnn
7401             { Others } { Boundary } { Default } { Boundary }
7402         }
7403         \tl_if_blank:eT
7404         { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
7405         {
7406             \xeCJK_copy_inter_class_toks:nnnn
7407             { Boundary } { Others } { Boundary } { Default }
7408         }
7409     }
7410 }

```

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

```

\__xeCJK_inactive_group_end:
7411 \cs_new_protected:Npn \__xeCJK_inactive_group_begin:
7412 { \group_begin: \makeXeCJKinactive }
7413 \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。

```

7414 \prop_const_from_keyval:Nn \c_xeCJK_ambiguous_char_prop
7415 {
7416     "00B7 = \textperiodcentered \textcentereddot \textcdot ,
7417     "2013 = \textendash ,
7418     "2014 = \textemdash ,

```

```

7419 "2018 = \textquoteleft \textgrq ,
7420 "2019 = \textquoteright ,
7421 "201C = \textquotedblleft \textgrqq ,
7422 "201D = \textquotedblright ,
7423 "2025 = \texthdofor ,
7424 "2026 = \textellipsis ,
7425 "2027 = \texthyphenationpoint ,
7426 "2E3A = \texttwoemdash
7427 }
7428 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_text_command: }
7429 \cs_new_protected:Npn \__xeCJK_patch_text_command:
7430 {
7431   \xeCJK_if_package_loaded:nTF { xunicode }
7432   { \__xeCJK_patch_xunicode_ambiguous_char: }
7433   {
7434     \exp_args:Ne \__xeCJK_patch_tuenc_ambiguous_char:n
7435     { \UnicodeEncodingName }
7436     \__xeCJK_patch_tuenc_accent:
7437     \__xeCJK_patch_tuenc_composite:
7438   }
7439 }
7440 \cs_new_protected:Npn \__xeCJK_patch_xunicode_ambiguous_char:
7441 {
7442   \RequirePackage { xunicode-addon }
7443   \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
7444   {
7445     \tl_map_inline:nn { ##2 }
7446     {
7447       \xunadd_set_begin_hook:nn { ####1 }
7448       { \__xeCJK_inactive_group_begin: }
7449       \xunadd_set_end_hook:nn { ####1 }
7450       { \__xeCJK_inactive_group_end: }
7451     }
7452   }
7453   \xunadd_append_begin_hook:n { \xeCJK_make_boundary: }
7454 }
7455 \cs_new_protected:Npn \__xeCJK_patch_tuenc_ambiguous_char:n #1
7456 {
7457   \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
7458   {
7459     \tl_map_inline:nn { ##2 }
7460     {
7461       \cs_if_exist:NF ####1
7462       { \DeclareTextSymbol ####1 {#1} { ##1 } }
7463       \__xeCJK_patch_ambiguous_char:nN {#1} ####1
7464     }
7465   }
7466 }
7467 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nN #1#2
7468 {
7469   \exp_args:Ne \__xeCJK_patch_ambiguous_char:nn
7470   { #1 \token_to_str:N #2 }
7471   { #1 - #2 }
7472 }
7473 \cs_new_protected:Npe \__xeCJK_patch_ambiguous_char:nNn #1#2#3
7474 {
7475   \exp_not:N \exp_args:Ne
7476   \exp_not:N \__xeCJK_patch_ambiguous_char:nn
7477   {
7478     \c_backslash_str #1
7479     \exp_not:N \token_to_str:N #2 -
7480     \exp_not:N \token_to_str:N #3
7481   }
7482   { #1 - #2#3 }
7483 }
7484 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nn #1#2
7485 {

```

```

7486 \cs_if_free:cF {#1}
7487 { \exp_args:Nc \__xeCJK_patch_ambiguous_char:Nn {#1} {#2} }
7488 }
7489 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:Nn #1#2
7490 {
7491   \token_if_chardef:NTF #1
7492   {
7493     \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
7494     { \int_eval:n {#1} }
7495     \cs_set_protected:Npe #1
7496     { \__xeCJK_ambiguous_char:n { \tex_Uchar:D #1 } }
7497   }
7498   {
7499     \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
7500     { \int_eval:n { \exp_after:wN ` #1 } }
7501     \cs_set_protected:Npe #1
7502     { \__xeCJK_ambiguous_char:n { \exp_not:o {#1} } }
7503   }
7504 }
7505 \cs_new_protected:Npn \__xeCJK_ambiguous_char:n #1
7506 {
7507   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7508   { \__xeCJK_inactive_group_begin: #1 \__xeCJK_inactive_group_end: }
7509   {#1}
7510 }
7511 \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

```

7512 \cs_new_protected:Npn \xeCJK_text_composite_patch:
7513 {
7514   \str_if_eq:eeT { \f@encoding } { \UnicodeEncodingName }
7515   { \xeCJK_make_boundary: }
7516 }

```

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。

```

7517 \cs_new_protected:Npn \__xeCJK_patch_tuenc_composite:
7518 {
7519   \cs_set_nopar:Npn \@text@composite@x
7520   {
7521     \xeCJK_text_composite_patch:
7522     \cs_if_exist_use:NF
7523   }
7524 }

```

`\__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。

```

7525 \group_begin:
7526 \char_set_catcode_other:n { "A0 }
7527 \cs_new_protected:Npn \__xeCJK_patch_tuenc_accent:
7528 {
7529   \cs_set_protected_nopar:Npn \add@unicode@accent ##1 ##2
7530   {
7531     \tl_if_blank:nTF { ##2 } { ^^a0 } { ##2 }
7532     \tex_Uchar:D \tex_numexpr:D ##1 \scan_stop:
7533   }
7534 }
7535 \group_end:

```

`\__xeCJK_patch_middle_dot:` 常被用作中文间隔号的 U+00B7 与 T1 等旧字体编码下定义的符号命令冲突。在 `encguide.pdf` 的编码符号表中, 如下定义有冲突。

```

\DeclareTextComposite{\r}{T1}{u}{183}
\DeclareTextSymbol{\cyrchvcrs}{T2A}{183}
\DeclareTextSymbol{\cyrchldsc}{T2B}{183}
\DeclareTextSymbol{\cyrabhha}{T2C}{183}
\DeclareTextSymbol\textvibyy{T3}{183}
\DeclareTextComposite{\B}{T4}{t}{183}
\DeclareTextComposite{\`}{T5}{\ecircumflex}{183}
\DeclareTextDoubleComposite{\`}{T5}{\^}{e}{183}
\DeclareTextSymbol{\textperiodcentered}{TS1}{183}
\DeclareTextSymbol{\cyrchldsc}{X2}{183}
\DeclareTextSymbol{\textperiodcentered}{LY1}{183}

```

LGR 编码的符号表有 183 号字符,但在 lgrenc.def 中未找到相应的符号命令。

```

7536 \prop_const_from_keyval:Nn \c__xeCJK_middle_dot_prop
7537 {
7538   T2A = \cyrchvcrs ,
7539   T2B = \cyrchldsc ,
7540   T2C = \cyrabhha ,
7541   X2  = \cyrchldsc ,
7542   TS1 = \textperiodcentered ,
7543   LY1 = \textperiodcentered ,
7544   T1  = \r u ,
7545   T4  = \B t ,
7546   T5  = \` \ecircumflex
7547 }
7548 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_middle_dot: }
7549 \cs_new_protected:Npn \__xeCJK_patch_middle_dot:
7550 {
7551   \prop_map_inline:Nn \c__xeCJK_middle_dot_prop
7552   { \__xeCJK_patch_middle_dot:nw { ##1 } ##2 \q_stop }
7553   \__xeCJK_patch_ambiguous_char:nNn { T5 } \` { \^ - e }
7554 }
7555 \cs_new_protected:Npn \__xeCJK_patch_middle_dot:nw #1#2#3 \q_stop
7556 {
7557   \tl_if_empty:nTF {#3}
7558   { \__xeCJK_patch_ambiguous_char:nN {#1} #2 }
7559   { \__xeCJK_patch_ambiguous_char:nNn {#1} #2 {#3} }
7560 }

```

pifont 宏包的符号\ding{183} 也有冲突。

```

7561 \__xeCJK_package_hook:nn { pifont }
7562 {
7563   \RenewDocumentCommand \Pifont { m }
7564   { \mode_leave_vertical: \makeXeCJKinactive \usefont { U } {#1} { m } { n } }
7565 }

```

\xeCJKchar \char 是 T_EX 原语,用户使用它直接通过字符编码从当前字体取字形,不经过 NFSS 层。一些宏包(如 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 LuaT_EX-j_a 提供 \ltjalchar / \ltjjachar 新命令的思路,xeCJK 提供 \xeCJKchar 命令,在输出字符前临时关闭 XeTeXinterchartokenstate,保证输出的字符来自当前字体而不被 interchar 拦截。与 LuaT_EX-j_a 不同的是,XeTeX 的 interchar 机制工作在 token 层面,没有节点级回调,因此无法自动区分 \char 和直接字符输入。 \xeCJKchar 供用户和第三方宏包显式使用,\char 本身不做任何修改。

用法与 \char 完全相同:\xeCJKchar"00B7 或 \xeCJKchar183。用户指南和手动补丁方法参见“已知问题和兼容性”一节。

```

7566 \cs_new_protected:Npn \xeCJK_default_char:w
7567 {
7568   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
7569   {
7570     \c_group_begin_token
7571     \tex_afterassignment:D \__xeCJK_default_char_aux:

```

```

7572         \l__xeCJK_default_char_int =
7573     }
7574     { \tex_char:D }
7575 }
7576 \cs_new_protected:Npn \__xeCJK_default_char_aux:
7577 {
7578     \tex_XeTeXinterchartokenstate:D = \c_zero_int
7579     \tex_char:D \l__xeCJK_default_char_int
7580     \c_group_end_token
7581 }
7582 \int_new:N \l__xeCJK_default_char_int
7583 \cs_new_eq:NN \xeCJKchar \xeCJK_default_char:w

```

mtpro2 的 `\overbrace` 和 `\underbrace` 通过 `\char` 访问 mt2exe 等字体的特殊位置,其中部分位置(如 183 = U+00B7)与 xeCJK 的 `interchar` 字符分类冲突。由于这两个命令只用在数学模式中,参数不含 CJK 字符,可以在入口处直接关闭 `XeTeXinterchartokenstate`,由数学模式的分组自动恢复。

```

7584 \__xeCJK_package_hook:nn { mtpo2 }
7585 {
7586     \cs_set_eq:NN \__xeCJK_mtpo_overbrace:w \overbrace
7587     \cs_gset_protected:Npn \overbrace #1
7588     {
7589         \makexeCJKinactive
7590         \__xeCJK_mtpo_overbrace:w {#1}
7591     }
7592     \cs_set_eq:NN \__xeCJK_mtpo_underbrace:w \underbrace
7593     \cs_gset_protected:Npn \underbrace #1
7594     {
7595         \makexeCJKinactive
7596         \__xeCJK_mtpo_underbrace:w {#1}
7597     }
7598 }

```

`\__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:
7599 \__xeCJK_package_hook:nn { unicode-math }
7600 {
7601     \prop_const_from_keyval:Nn \c__xeCJK_um_ambiguous_char_prop
7602     {
7603         "00B7 = \cdotp ,
7604         "2025 = \enleadertwodots ,
7605         "2026 = \unicodeellipsis
7606     }
7607     \cs_new_protected:Npn \__xeCJK_save_um_char:
7608     {
7609         \cs_set_protected:Npe \__xeCJK_restore_um_char:
7610         {
7611             \prop_map_function:NN
7612             \c__xeCJK_um_ambiguous_char_prop
7613             \__xeCJK_restore_um_char_aux:nn
7614         }
7615     }
7616     \cs_new_eq:NN \__xeCJK_restore_um_char: \prg_do_nothing:
7617     \cs_new:Npn \__xeCJK_restore_um_char_aux:nn #1#2
7618     {
7619         \__xeCJK_gset_mathcodenum:nn
7620         { \int_value:w #1 }
7621         { \int_value:w \tex_Umathcodenum:D #1 }
7622     }
7623     \cs_new_protected:Npn \__xeCJK_gset_mathcodenum:nn #1#2
7624     {
7625         \int_compare:nNnF { \tex_Umathcodenum:D #1 } = {#2}
7626         { \tex_global:D \tex_Umathcodenum:D #1 = #2 ~ }
7627     }
7628 }

```

`\_xeCJK_patch_microtype_get_slot:` 兼容 microtype。

```

7629 \cs_new_protected:Npn \_xeCJK_patch_microtype_get_slot:
7630 {
7631   \cs_new_eq:NN \xeCJK@original@get@slot \MT@get@slot@
7632   \cs_set_eq:NN \MT@get@slot@ \xeCJK@microtype@get@slot
7633   \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
7634 }
7635 \cs_new_protected_nopar:Npn \xeCJK@microtype@get@slot
7636 {
7637   \int_compare:nNt \MT@char < \c_zero_int
7638   { \_xeCJK_get_ambiguous_slot: }
7639   \xeCJK@original@get@slot
7640 }
7641 \cs_new_protected:Npn \_xeCJK_get_ambiguous_slot:
7642 {
7643   \prop_get:NeNT \g__xeCJK_ambiguous_slot_prop
7644   { \MT@encoding - \tex_the:D \MT@toks } \l__xeCJK_tmp_tl
7645   { \cs_set_eq:NN \MT@char \l__xeCJK_tmp_tl }
7646 }
7647 \cs_new_protected:Npn \xeCJK@microtype@restore@pickupfont
7648 { \_xeCJK_gadd_font_initial_hook:n { \MT@ltx@pickupfont } }
7649 \_xeCJK_package_hook:nn { microtype }
7650 {
7651   \cs_if_free:NF \MT@get@slot@
7652   { \_xeCJK_patch_microtype_get_slot: }
7653   \MT@addto@setup { \xeCJK@microtype@restore@pickupfont }
7654 }

```

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 `\CJKe glue` 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` `whatsit`。
- `\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>}}{<content>}`。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 `\CJKe glue` (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 38.33002 pt, 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, `hxeTeX`、`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。

```

7655 \cs_new_protected:Npn \__xeCJK_boundary_register_hyperref_annot:
7656 {
7657   \clist_map_inline:nn
7658   {
7659     Hy@BeginAnnot , Hy@EndAnnot , hyper@link , hyper@linkurl ,
7660     href , hyperref
7661   }
7662   { \__xeCJK_boundary_reserve:n {##1} }
7663   \cs_gset_eq:NN \__xeCJK_orig_Hy_BeginAnnot:n \Hy@BeginAnnot
7664   \cs_gset_eq:NN \__xeCJK_orig_Hy_EndAnnot: \Hy@EndAnnot
7665   \cs_gset_protected:Npn \Hy@BeginAnnot ##1
7666   {
7667     \__xeCJK_boundary_inline_stream_begin:n { auto }
7668     \__xeCJK_orig_Hy_BeginAnnot:n { ##1 }
7669   }
7670   \cs_gset_protected:Npn \Hy@EndAnnot
7671   {
7672     \int_compare:nNtT { \c@Hy@AnnotLevel } = { 1 }
7673     {
7674       \__xeCJK_if_last_math:T
7675       {
7676         \tl_if_eq:cnF
7677         {
7678           g__xeCJK_boundary_capture_
7679           \int_use:N \g__xeCJK_boundary_capture_depth_int _last_tl
7680         }
7681         { math }
7682         { \__xeCJK_boundary_capture_class:n { default } }
7683       }
7684     }
7685     \__xeCJK_orig_Hy_EndAnnot:
7686     \__xeCJK_boundary_inline_stream_end:n { auto }
7687   }

```

```

7688 \cs_gset_eq:NN \__xeCJK_orig_hyper_link:nnn \hyper@link
7689 \cs_gset_protected:Npn \hyper@link ##1##2##3
7690 {
7691   \__xeCJK_orig_hyper_link:nnn {##1} {##2}
7692   {
7693     \__xeCJK_boundary_math_begin:n {##3}
7694     ##3
7695     \__xeCJK_boundary_math_end:n {##3}
7696   }
7697 }
7698 \cs_gset_eq:NN \__xeCJK_orig_hyper_linkurl:nn \hyper@linkurl
7699 \cs_gset_protected:Npn \hyper@linkurl ##1##2
7700 {
7701   \__xeCJK_orig_hyper_linkurl:nn
7702   {
7703     \__xeCJK_boundary_math_begin:n {##1}
7704     ##1
7705     \__xeCJK_boundary_math_end:n {##1}
7706   }
7707   {##2}
7708 }
7709 \__xeCJK_boundary_register_transparent:n { Hy@raisedlink }
7710 \cs_if_exist:NT \hyper@anchor
7711 { \__xeCJK_boundary_register_transparent:n { hyper@anchor } }
7712 \cs_if_exist:NT \__hyp_target_raise:n
7713 {
7714   \__xeCJK_boundary_wrap_transparent_onearg_braced:NN
7715   \__xeCJK_orig_hyp_target_raise:n \__hyp_target_raise:n
7716 }
7717 }
7718 \__xeCJK_package_hook:nn { hyperref }
7719 {
7720   \pdfstringdefDisableCommands
7721   {
7722     \__xeCJK_gobble_CJKfamily:
7723     \xeCJK_cs_clear:N \__xeCJK_inactive_group_begin:
7724     \xeCJK_cs_clear:N \__xeCJK_inactive_group_end:
7725     \xeCJK_cs_clear:N \makexeCJKinactive
7726     \xeCJK_cs_clear:N \xeCJK_text_composite_patch:
7727   }
7728   \__xeCJK_boundary_register_hyperref_annot:
7729 }

```

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。

```

7730 % \tn{colorbox} 和 \tn{fcolorbox} 的扫描器都会把可见正文留作
7731 % \tn{color@b@x} 的第三个参数。先包装这个共同入口, 再安装 wrapped-box
7732 % 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。
7733 % \pkg{xcolor} 会重定义这个私有函数, 因此它的 package hook 需要按
7734 % 当前定义重新保存原函数。
7735 \cs_new_protected:Npn \__xeCJK_boundary_color_box:nnn #1#2#3
7736 {
7737   \__xeCJK_boundary_math_begin:n {#3}
7738   \__xeCJK_orig_color_box:nnn {#1} {#2}
7739   { #3 \__xeCJK_boundary_math_end:n {#3} }
7740 }
7741 \cs_new_protected:Npn \__xeCJK_boundary_prepare_color_box:
7742 {
7743   \__xeCJK_boundary_reserve:n { color@b@x }
7744   \cs_if_eq:NNF \color@b@x \__xeCJK_boundary_color_box:nnn
7745   {

```

```

7746     \cs_gset_eq:NN \__xeCJK_orig_color_box:nnn \color@b@x
7747     \cs_gset_eq:NN \color@b@x \__xeCJK_boundary_color_box:nnn
7748   }
7749 }
7750 \cs_new_protected:Npn \__xeCJK_boundary_textcolor:nnn #1#2#3
7751 {
7752   \__xeCJK_boundary_if_math_edge:nTF {#3}
7753   {
7754     \__xeCJK_boundary_inline_stream_begin:n { auto }
7755     \__xeCJK_boundary_math_begin:n {#3}
7756     \__xeCJK_orig_textcolor:nnn {#1} {#2}
7757     { #3 \__xeCJK_boundary_math_end:n {#3} }
7758     \__xeCJK_boundary_inline_stream_end:n { auto }
7759   }
7760   { \__xeCJK_orig_textcolor:nnn {#1} {#2} {#3} }
7761 }
7762 \cs_new_protected:Npn \__xeCJK_boundary_register_textcolor:
7763 {
7764   \__xeCJK_boundary_reserve:n { @textcolor }
7765   \__xeCJK_boundary_reserve:n { textcolor }
7766   \cs_if_eq:NNF \@textcolor \__xeCJK_boundary_textcolor:nnn
7767   {
7768     \cs_gset_eq:NN \__xeCJK_orig_textcolor:nnn \@textcolor
7769     \cs_gset_eq:NN \@textcolor \__xeCJK_boundary_textcolor:nnn
7770   }
7771 }
7772 \cs_new_protected:Npn \__xeCJK_boundary_register_color:
7773 {
7774   \__xeCJK_boundary_reserve:n { colorbox }
7775   \__xeCJK_boundary_reserve:n { fcolorbox }
7776   \__xeCJK_boundary_register_transparent:n { set@color }
7777   \__xeCJK_boundary_register_transparent:n { reset@color }
7778   \__xeCJK_boundary_prepare_color_box:
7779   \__xeCJK_boundary_register_wrapped_box:nn { color@b@x } { auto }
7780   \__xeCJK_boundary_register_textcolor:
7781 }
7782 \__xeCJK_package_hook:nn { color }
7783 { \__xeCJK_boundary_register_color: }
7784 \__xeCJK_package_hook:nn { xcolor }
7785 { \__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。

```

7786 \cs_new_protected:Npn \__xeCJK_boundary_register_l3color:
7787 {
7788   \__xeCJK_boundary_wrap_transparent_onearg:NN
7789   \__xeCJK_orig_color_select:N \__color_select:N
7790   \__xeCJK_boundary_wrap_transparent_noarg:NN
7791   \__xeCJK_orig_color_backend_reset: \__color_backend_reset:
7792 }
7793 \__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。

```

7794 \__xeCJK_package_hook:nn { hypdoc }
7795 { \__xeCJK_boundary_register_transparent:n { HD@target } }

```

siunitx 用 math 模式排版数字与单位。math 不触发 interchar 转换, 入口的 \mathon 又盖住左侧 marker, 情况与修复前的 \eqref 相同; 数字与单位输出必然以西文或数字开始、结束, 因此也注册为固定 Default 首尾的 stream capture。v2 旧名 \si、\SI 是独立顶层命令, 不经由 \unit、\qty 的 cmd hook, 需单独注册。注册前先检查命令是否存在, 以兼容未来移除旧名的 siunitx 版本。若其他宏包恰好定义了同名 \si/\SI, 这项检查也会把它们一并注册——误注册只是多包一层 Default stream, 不改变无 CJK 邻接时的排版, 记录为已知限制。数学模式内使用(如 $x=\text{unit}\{\text{kg}\}$)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。

```

7796 \cs_new_protected:Npn \__xeCJK_boundary_register_siunitx:
7797 {
7798   \__xeCJK_boundary_register_stream:nn { unit } { default }
7799   \__xeCJK_boundary_register_stream:nn { qty } { default }
7800   \__xeCJK_boundary_register_stream:nn { num } { default }
7801   \cs_if_exist:NT \si
7802     { \__xeCJK_boundary_register_stream:nn { si } { default } }
7803   \cs_if_exist:NT \SI
7804     { \__xeCJK_boundary_register_stream:nn { SI } { default } }
7805 }
7806 \__xeCJK_package_hook:nn { siunitx }
7807 { \__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 Xe_{La}TeX 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。

```

7808 \cs_new_protected:Npn \__xeCJK_boundary_register_url:
7809 {
7810   \clist_map_inline:nn { url , path , nolinkurl , Url@z }
7811     { \__xeCJK_boundary_reserve:n {##1} }
7812   \cs_if_exist:NF \__xeCJK_orig_Url_z:n
7813     {
7814       \cs_set_eq:NN \__xeCJK_orig_Url_z:n \Url@z
7815       \cs_gset_protected:Npn \Url@z ##1
7816         {
7817           \__xeCJK_boundary_inline_stream_begin:n { default }
7818           \__xeCJK_orig_Url_z:n {##1}
7819           \__xeCJK_boundary_inline_stream_end:n { default }
7820         }
7821     }
7822 }
7823 \__xeCJK_package_hook:nn { url } { \__xeCJK_boundary_register_url: }

```

LaTeX 2_ε 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; shortverb 的短记号与 \verb* 共用同一入口和组结束点, 因此继续自动覆盖。

补丁仍在 `\_xeCJK_after_preamble:n` 安装, 以避开其他宏包在导言区重定义 `\verb` 的时序。

`\verb` 右侧还有一处独立时序(#919): `\verb` 组内把 `\language` 切到 `\l@nohyphenation`, 组关闭后的 `language whatsit` 会延迟到下一字符进入列表时才排出。若让它留到 `capture` 结束之后, 它会遮住 `framework` 刚刚重放的末尾 `marker`。这里仍在 `\verb@egroup` 关组后立即执行 `\setlanguage\language`, 把 `language whatsit` 收进 `stream capture`; 随后 `framework` 在它之后重放实际末类别, 右侧有、无源码空格都交给普通恢复链处理。旧的通用 `whatsit` 类别猜测已经删除, 不再参与这条路径。

`\_xeCJK_flush_language_whatsit:` 按当前 `\language` 值主动排出一个 `language whatsit`。`\setlanguage\language` 只排出 `whatsit` 记录当前值, 不改变 `hyphenation` 状态, 对断行无副作用。仅在外层水平模式下执行: 数学模式或行首场景没有上述时序问题, `restricted horizontal mode` (`\hbox` 内) 则根本不产生延迟 `whatsit`。目前唯一调用点是 `\verb@egroup` (见下方 `\_xeCJK_boundary_register_verb:`); 未来若有其他切换 `\language` 的分组宏需要同样的时序修正, 可直接复用本函数, 并在此注释中登记新的调用点。

```
7824 \cs_new_protected:Npn \_xeCJK_flush_language_whatsit:
7825   { \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: 7826 \cs_new_protected:Npn \_xeCJK_boundary_register_verb:
7827   {
7828     \cs_if_exist:NF \_xeCJK_orig_verb:
7829     {
7830       \cs_set_eq:NN \_xeCJK_orig_verb: \verb
7831       \cs_gset_protected:Npn \verb
7832       {
7833         \_xeCJK_boundary_inline_stream_begin:n { auto }
7834         \_xeCJK_orig_verb:
7835       }
7836       \tl_gput_right:Nn \verb@egroup
7837       {
7838         \_xeCJK_flush_language_whatsit:
7839         \_xeCJK_boundary_inline_stream_end:n { auto }
7840       }
7841     }
7842   }
7843 \_xeCJK_boundary_reserve:n { verb }
7844 \_xeCJK_boundary_reserve:n { verb@egroup }
7845 \_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 `XYTeX` 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 \CJKeclue(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: 7846 \cs_new_protected:Npn \__xeCJK_boundary_register_codedoc_meta:
7847 {
7848   \cs_if_exist:NTF \__codedoc_meta:n
7849   {
7850     \clist_map_inline:nn
7851       { \__codedoc_meta:n , Arg , marg , oarg , par }
7852       { \__xeCJK_boundary_reserve:n {##1} }
7853     \cs_if_exist:NF \__xeCJK_orig_codedoc_meta:n
7854     {
7855       \cs_set_eq:NN \__xeCJK_orig_codedoc_meta:n \__codedoc_meta:n
7856       \cs_gset_protected:Npn \__codedoc_meta:n ##1
7857         { \__xeCJK_orig_codedoc_meta:n { \hbox:n {##1} } }
7858       \__xeCJK_boundary_register_stream:nn { meta } { default }
7859     }
7860   }
7861   {
7862     \bool_lazy_and:nnT
7863       { \cs_if_exist_p:c { ver@doc.sty } }
7864       { \cs_if_free_p:N \__xeCJK_orig_codedoc_meta:n }
7865     {
7866       \__xeCJK_boundary_reserve:n { meta }
7867       \cs_set_eq:NN \__xeCJK_orig_codedoc_meta:n \meta
7868       \DeclareDocumentCommand \meta { m }
7869       {
7870         \__xeCJK_boundary_inline_stream_begin:n { default }
7871         \__xeCJK_orig_codedoc_meta:n { \hbox:n {##1} }
7872         \__xeCJK_boundary_inline_stream_end:n { default }
7873       }
7874     }
7875   }
7876 }
7877 \__xeCJK_at_end_preamble:n { \__xeCJK_boundary_register_codedoc_meta: }
7878 % \changes{v3.10.4}{2026/07/20}{将 \tn{eqref} 注册为固定 Default
7879 % 首尾、将 \tn{cs} 注册为仅固定 Default 首端的 stream capture;
7880 % codedoc/doc 的 meta 内部适配器直接包围固定 Default stream, 删除专用
7881 % drain 并覆盖其所有公共调用方 (\#992)。}
7882 % \changes{v3.10.4}{2026/07/20}{将内核 \tn{@setref} (或 \pkg{hyperref}
7883 % 保存的 \tn{real@setref}) 注册为 auto stream capture, 以统一框架取代
7884 % \#991 的专用 saved-node/replay 补丁。}
7885 % \tn{eqref} 的圆括号和 \tn{meta} 的数学尖括号决定了命令两端
7886 % 始终是 Default; 参数内部即使包含中文, 也不能把外侧 oracle
7887 % 误判为裸中文。 \tn{cs} 只有开头的反斜线固定为 Default, 末尾仍由
7888 % 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
7889 % 真实末类别。
7890 \__xeCJK_at_end_preamble:n
7891 {
7892   \cs_if_exist:NTF \real@setref
7893     { \__xeCJK_boundary_register_stream:nn { real@setref } { auto } }
7894     { \__xeCJK_boundary_register_stream:nn { @setref } { auto } }
7895   \cs_if_exist:NT \eqref
7896     { \__xeCJK_boundary_register_stream:nn { eqref } { default } }
7897   \cs_if_exist:NT \cs
```

```

7898     { \_xeCJK_boundary_register_stream:nn { cs } { first-default } }
7899 }

```

`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: 7900 \cs_new_protected:Npn \_xeCJK_boundary_register_biblatex_pagetracker:
7901 {
7902   \cs_if_exist:NT \blx@pagetracker
7903   { \_xeCJK_boundary_register_transparent:n { blx@pagetracker } }
7904 }
7905 \_xeCJK_at_end_preamble:n { \_xeCJK_boundary_register_biblatex_pagetracker: }
7906 % 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
7907 % 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,
7908 % 避免不存在的控制序列被 LaTeX 静默跳过。
7909 \_xeCJK_at_end_preamble:n { \_xeCJK_boundary_user_register_apply: }
7910 \_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。

```

7911 \_xeCJK_package_hook:nn { cprotect }
7912 {
7913   \cs_if_free:NF \icprotect
7914   { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
7915 }

```

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

```

7916 \_xeCJK_package_hook:nn { listings }
7917 { \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。

```

7918 \_xeCJK_package_hook:nn { everyysel }
7919 {
7920   \cs_if_exist:NF \@EverySelectfont@Legacy
7921   { \cs_undefine:c { ver@CJK . \c__xeCJK_package_ext_tl } }
7922 }

```

`\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。

```

7923 \ctex_at_begin_package:nn { CJKnumb }
7924 {
7925   \tl_new:N \l__xeCJK_CJK_version_tl
7926   \tl_set_eq:Nc \l__xeCJK_CJK_version_tl { ver@CJK . \c__xeCJK_package_ext_tl }

```

```

7927 \tl_set:cn { ver@CJK . \c__xeCJK_package_ext_tl } { 9999/99/99 }
7928 \cs_new_protected:Npn \CJKaddEncHook #1#2
7929 {
7930   \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
7931   {
7932     \group_begin:
7933     \cs_set_eq:NN \Unicode \xeCJK_unicode_char:nn
7934     \cs_set_eq:NN \def \xdef
7935     #2
7936     \group_end:
7937     \str_gset:Nn \CJK@tenthousand { ~~~~4e07 }
7938     \str_gset:Nn \CJK@hundredmillion { ~~~~4ebf }
7939     \tl_if_exist:NF \CJK@UnicodeEnc
7940     { \tl_const:Nn \CJK@UnicodeEnc { UTF8 } }
7941     \cs_if_exist:NF \Unicode
7942     { \cs_new_eq:NN \Unicode \xeCJK_unicode_char:nn }
7943   }
7944 }
7945 \cs_new:Npn \xeCJK_unicode_char:nn #1#2
7946 { \tex_Uchar:D \tex_numexpr:D (#1) * 256 + (#2) \scan_stop: }
7947 }
7948 \ctex_at_end_package:nn { CJKnumb }
7949 { \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。

```

7950 \bool_if:NT \g__xeCJK_config_bool
7951 {
7952   \ExplSyntaxOff
7953   \file_input:n { \g__xeCJK_config_name_tl .cfg }
7954   \ExplSyntaxOn
7955 }
7956 </package>

```

5.19 xeCJKfntef

```

7957 <fntef>

7958 \PassOptionsToPackage { normalem } { ulem }
7959 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
7960 \ProcessOptions \scan_stop:

7961 \RequirePackage { xeCJK }
7962 \RequirePackage { ulem }
7963 \RequirePackage { l3draw }
7964 \RequirePackage { accsupp }

7965 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

\xeCJK_hook_for_ulem: 7966 \cs_new_protected:Npn \xeCJK_hook_for_ulem:
7967 {
7968   \xeCJK_ulem_detect_node:
7969   \l__xeCJK_ulem_text_format_tl
7970   \bool_if:NF \l__xeCJK_ulem_hook_used_bool
7971   {
7972     \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
7973     \__xeCJK_ulem_hook:
7974   }
7975   \xeCJK_ulem_begin_node:
7976 }
7977 \cs_new_protected:Npn \__xeCJK_ulem_hook:
7978 {
7979   \__xeCJK_ulem_initial:
7980   \bool_if:NTF \l__xeCJK_ulem_periodic_bool
7981   {
7982     \seq_gput_right:Nn \g__xeCJK_ulem_periodic_later_seq { 0 }
7983     \cs_set_eq:NN \UL@spfactor \__xeCJK_ulem_spfactor_pending:
7984     \xeCJK_swap_cs:NN

```

```

7985         \UL@leaders \xeCJK_ulem_periodic_leaders:
7986         \cs_set_eq:NN \__xeCJK_ulem_periodic_var_leaders:
7987         \xeCJK_ulem_periodic_var_leaders:
7988         \cs_set_eq:NN \xeCJK_ulem_right_skip:
7989         \__xeCJK_ulem_periodic_right_skip:
7990     }
7991     {
7992         \bool_if:NT \l__xeCJK_ulem_subtract_bool
7993         {
7994             \xeCJK_swap_cs:NN \UL@leaders \xeCJK_ulem_leaders:
7995             \cs_set_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:
7996             \cs_set_eq:NN \xeCJK_ulem_right_skip: \__xeCJK_ulem_right_skip:
7997         }
7998     }
7999     \bool_if:NT \l__xeCJK_ulem_hidden_bool
8000     { \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_hidden_box: }
8001     \bool_if:NTF \l__xeCJK_ulem_skip_bool
8002     {
8003         \cs_set_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
8004         \cs_set_eq:NN \__xeCJK_ulem_hskip_aux:n \xeCJK_ulem_hskip:n
8005     }
8006     {
8007         \xeCJK_swap_cs:NN \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
8008         \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
8009         \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
8010     }
8011     \xeCJK_glue_to_skip:nN
8012     {
8013         \cs_set_eq:NN \ \tex_space:D
8014         \cs_set_eq:NN \penalty \tex_penalty:D
8015         \cs_set_eq:NN \hskip \skip_horizontal:N
8016         \CJKglue
8017     } \l__xeCJK_ccglue_skip
8018     \xeCJK_glue_to_skip:nN
8019     {
8020         \cs_set_eq:NN \ \tex_space:D
8021         \cs_set_eq:NN \penalty \tex_penalty:D
8022         \cs_set_eq:NN \hskip \skip_horizontal:N
8023         \CJKecglue
8024     } \l__xeCJK_ecglue_skip
8025     \xeCJK_glue_to_skip:nN { \xeCJK_space_glue: } \l__xeCJK_space_skip
8026     \cs_set_protected:Npn \CJKglue
8027     { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
8028     \cs_set_protected:Npn \CJKecglue
8029     { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
8030     \cs_set_protected:Npn \xeCJK_space_glue:
8031     { \__xeCJK_ulem_glue:n \l__xeCJK_space_skip }
8032     \cs_set_eq:NN \xeCJK_punct_node:N \use_none:n
8033     \cs_set_eq:NN \xeCJK_if_last_punct:TF \use_ii:nn
8034     \keys_set:nn { xeCJK / options }
8035     { CheckFullRight = false , xCJKecglue = false }
8036     }
8037     \skip_new:N \l__xeCJK_space_skip
8038     \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

```

8039 \clist_map_inline:nn
8040 {
8041     UL@word , ULon , UL@on , UL@onin ,
8042     uuline , uuline , uwave , sout , xout , dashuline , dotuline ,
8043     xeCJKfntefon , CJKunderline , CJKunderwave , CJKunderdblline ,
8044     CJKsout , CJKxout , CJKunderanyline , CJKunderanysymbol , CJKunderdot
8045 }
8046 { \__xeCJK_boundary_reserve:n {#1} }
8047 \cs_new_protected:Npn \xeCJK_ulem_word:nw #1 ~
8048 {
8049     \__xeCJK_ulem_start:w #1 ~

```

```

8050 \exp_after:wN \if_meaning:w \exp_after:wN \UL@end #1
8051 \exp_after:wN \_xeCJK_ulem_end:
8052 \else:
8053 \exp_after:wN \_xeCJK_ulem_loop:nw
8054 \fi:
8055 }
8056 \cs_new_protected:Npn \_xeCJK_ulem_end:
8057 {
8058 \c_group_end_token
8059 \c_group_end_token
8060 \tex_unskip:D \tex_unskip:D \tex_unskip:D
8061 \xeCJK_ulem_right_skip:
8062 \xeCJK_ulem_group_end:
8063 \xeCJK_ulem_right_node:
8064 \_xeCJK_boundary_inline_stream_end:n { auto }
8065 \bool_set_false:N \l__xeCJK_ulem_stream_started_bool
8066 \skip_gzero:N \g__xeCJK_ulem_pending_shrink_skip
8067 \int_set:Nn \tex_spacefactor:D { \UL@spfactor }
8068 }
8069 \cs_new_protected:Npn \_xeCJK_ulem_loop:nw
8070 {
8071 \reverse_if:N \if_mode_math:
8072 \reverse_if:N \if_dim:w \tex_lastskip:D = \c_zero_dim
8073 \skip_gset_eq:NN \UL@skip \tex_lastskip:D
8074 \tex_unskip:D
8075 \UL@stop
8076 \_xeCJK_ulem_flush_pending_shrink:
8077 \UL@leaders
8078 \fi:
8079 \fi:
8080 \xeCJK_ulem_word:nw \prg_do_nothing:
8081 }
8082 \cs_new_protected:Npn \_xeCJK_ulem_start:w
8083 { \exp_after:wN \UL@start }
8084 \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:

```

8085 \cs_new_protected:Npn \xeCJK_ulem_left:
8086 {
8087 \xeCJK_ulem_left_node:
8088 \xeCJK_make_group_tag:
8089 }
8090 \cs_new_eq:NN \xeCJK_ulem_left_node: \prg_do_nothing:
8091 \cs_new_protected:Npn \xeCJK_ulem_detect_node:
8092 {
8093 \scan_stop:
8094 \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
8095 {
8096 \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
8097 \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
8098 \cs_set_eq:NN \_xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8099 }
8100 {
8101 \dim_set_eq:NN \l__xeCJK_tmp_dim \tex_lastkern:D
8102 \tex_unkern:D
8103 \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_tmp_dim }
8104 {
8105 \tex_unkern:D
8106 \cs_set_protected:Npe \xeCJK_ulem_left_node:
8107 {
8108 \tex_kern:D - \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
8109 \tex_kern:D \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
8110 }
8111 \cs_set_protected:Npn \xeCJK_ulem_begin_node:
8112 { { \xeCJK_make_node:n { ulem-begin } } }
8113 \cs_set_eq:NN \_xeCJK_ulem_hskip:n \_xeCJK_ulem_hskip_first:n
8114 }

```

```

8115         {
8116             \tex_kern:D \l__xeCJK_tmp_dim
8117             \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
8118             \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
8119             \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8120         }
8121     }
8122 }
8123 \xeCJK_declare_node:n { ulem-begin }
8124 \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。

```

8125 \cs_new_protected:Npn \__xeCJK_ulem_hskip_first:n #1
8126 {
8127     \xeCJK_if_last_node:nTF { ulem-begin }
8128     {
8129         \xeCJK_remove_node:
8130         \skip_horizontal:n {#1}
8131     }
8132     { \xeCJK_ulem_hskip:n {#1} }
8133     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
8134 }
8135 \cs_new_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
8136 \cs_new_protected:Npn \xeCJK_ulem_hskip:n #1
8137 { { \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。

```

8138 \cs_new_protected:Npn \xeCJK_ulem_right:
8139 {
8140     \scan_stop:
8141     \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
8142     { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
8143     {
8144         \dim_compare:nNnTF \tex_lastkern:D = { 3sp }
8145         { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
8146         {
8147             \exp_args:NNo \tex_unkern:D
8148             \__xeCJK_ulem_right_aux:n { \dim_use:N \tex_lastkern:D }
8149         }
8150     }
8151 }
8152 \cs_new_protected:Npn \__xeCJK_ulem_right_aux:n #1
8153 {
8154     \dim_compare:nNnTF \tex_lastkern:D = { - #1 }
8155     {
8156         \tex_unkern:D
8157         \cs_gset_protected:Npn \xeCJK_ulem_right_node:
8158         {
8159             \tex_kern:D - #1 \exp_stop_f:
8160             \tex_kern:D #1 \exp_stop_f:
8161         }
8162         \tl_gset:Nx \UL@spfactor { \int_use:N \tex_spacefactor:D }
8163     }
8164     {
8165         \tex_kern:D #1 \exp_stop_f:
8166         \xeCJK_cs_gclear:N \xeCJK_ulem_right_node:
8167     }
8168 }
8169 \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。

```

8170 \cs_new_protected:Npn \xeCJK_ulem_leaders:
8171 { \__xeCJK_ulem_var_leaders: }
8172 \cs_new_protected:Npn \xeCJK_ulem_var_leaders:
8173 {
8174   \scan_stop:
8175   \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8176   {
8177     \skip_horizontal:N \UL@pixel
8178     \UL@leadtype \skip_horizontal:N \UL@skip
8179     \skip_horizontal:n { - \UL@pixel }
8180     \cs_gset_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_leaders:
8181   }
8182 }
8183 \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_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。

```

8184 \cs_new_protected:Npn \xeCJK_ulem_periodic_leaders:
8185 { \__xeCJK_ulem_periodic_var_leaders: }
8186 \cs_new_protected:Npn \xeCJK_ulem_periodic_var_leaders:
8187 {
8188   \scan_stop:
8189   \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8190   {
8191     \cs_if_eq:NNTF \UL@spfactor \__xeCJK_ulem_spfactor_pending:
8192     { \__xeCJK_ulem_periodic_first_leaders: }
8193     {
8194       \skip_if_eq:nnTF { \UL@skip } { \box_wd:N \UL@box }
8195       { \__xeCJK_ulem_periodic_first_leaders: }
8196       { \xeCJK_ulem_periodic_leaders: }
8197     }
8198   }
8199 }
8200 \cs_new_protected:Npn \__xeCJK_ulem_periodic_first_leaders:
8201 {
8202   \bool_if:NNTF \l__xeCJK_ulem_subtract_bool
8203   {
8204     \__xeCJK_ulem_periodic_clipped_leaders:nnn
8205     { \UL@skip } { \UL@pixel } { \UL@pixel }
8206   }
8207   {
8208     \__xeCJK_ulem_periodic_clipped_leaders:nnn
8209     { \UL@skip } { - \UL@pixel } { - \UL@pixel }
8210   }
8211   \cs_gset_eq:NN \__xeCJK_ulem_periodic_var_leaders:
8212   \__xeCJK_ulem_periodic_later_leaders:
8213 }
8214 \cs_new_eq:NN \__xeCJK_ulem_periodic_var_leaders:
8215 \xeCJK_ulem_periodic_var_leaders:
8216 \cs_new_protected:Npn \__xeCJK_ulem_periodic_later_leaders:
8217 {

```

```

8218 \scan_stop:
8219 \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
8220 {
8221   \cs_if_eq:NNTF \UL@spfactor \__xeCJK_ulem_spfactor_pending:
8222   {
8223     \seq_get_right:NN
8224     \g__xeCJK_ulem_periodic_later_seq \l__xeCJK_tmp_tl
8225     \tl_if_eq:NnTF \l__xeCJK_tmp_tl { 1 }
8226     { \xeCJK_ulem_periodic_leaders: }
8227     {
8228       \seq_gpop_right:NN
8229       \g__xeCJK_ulem_periodic_later_seq \l__xeCJK_tmp_tl
8230       \seq_gput_right:Nn \g__xeCJK_ulem_periodic_later_seq { 1 }
8231       \bool_if:NTF \l__xeCJK_ulem_subtract_bool
8232       {
8233         \__xeCJK_ulem_periodic_clipped_leaders:nnn
8234         { \c_zero_skip } { - \UL@pixel } { \c_zero_dim }
8235         \xeCJK_ulem_periodic_leaders:
8236         \__xeCJK_ulem_periodic_clipped_leaders:nnn
8237         { \c_zero_skip } { \c_zero_dim } { - \UL@pixel }
8238       }
8239       { \xeCJK_ulem_periodic_leaders: }
8240     }
8241   }
8242   { \xeCJK_ulem_periodic_leaders: }
8243 }
8244 }
8245 \cs_new_protected:Npn \__xeCJK_ulem_periodic_clipped_leaders:nnn #1#2#3
8246 {
8247   \skip_set:Nn \l__xeCJK_tmp_skip { #1 - #2 - #3 }
8248   \dim_compare:nNnTF { \l__xeCJK_tmp_skip } > { \c_zero_dim }
8249   {
8250     \tex_kern:D \dim_eval:n {#2}
8251     \tex_special:D { x:gsave }
8252     \tex_special:D
8253     {
8254       pdf:literal~
8255       0~
8256       \dim_to_decimal_in_bp:n { - \box_dp:N \ULC@box - 1pt } ~
8257       \dim_to_decimal_in_bp:n { \l__xeCJK_tmp_skip } ~
8258       \dim_to_decimal_in_bp:n
8259       { \box_ht:N \ULC@box + \box_dp:N \ULC@box + 2pt } ~
8260       re~W~n
8261     }
8262     \tex_kern:D \dim_eval:n { - 2 \UL@pixel }
8263     \xeCJK_no_break:
8264     \UL@leadtype
8265     \skip_horizontal:n { \l__xeCJK_tmp_skip + 4 \UL@pixel }
8266     \tex_kern:D \dim_eval:n { - 2 \UL@pixel }
8267     \tex_special:D { x:grestore }
8268     \tex_kern:D \dim_eval:n {#3}
8269   }
8270   { \skip_horizontal:n {#1} }
8271 }
8272 \bool_new:N \l__xeCJK_ulem_periodic_bool
8273 \seq_new:N \g__xeCJK_ulem_periodic_later_seq
8274 \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。

```

8275 \cs_new_eq:NN \xeCJK_ulem_right_skip: \prg_do_nothing:
8276 \cs_new_protected:Npn \__xeCJK_ulem_right_skip:
8277 {
8278   \int_case:nn { \tex_lastnodetype:D }
8279   {

```

```

8280     { \c__xeCJK_hlist_node } { \__xeCJK_ulem_right_skip_hbox: }
8281     { \c__xeCJK_glue_node } { \__xeCJK_ulem_right_skip_glue: }
8282     { \c__xeCJK_penalty_node } { \__xeCJK_ulem_right_skip_penalty: }
8283   }
8284 }
8285 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_hbox:
8286 {
8287   \box_set_to_last:N \l__xeCJK_tmp_box
8288   \__xeCJK_if_last_kern:TF
8289     { \__xeCJK_ulem_right_skip_kern: }
8290     { \__xeCJK_ulem_right_skip_glue: }
8291   \box_use_drop:N \l__xeCJK_tmp_box
8292 }
8293 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_kern:
8294 {
8295   \dim_set:Nn \l__xeCJK_tmp_dim { - \box_wd:N \l__xeCJK_tmp_box }
8296   \dim_compare:nNnT \tex_lastkern:D = \l__xeCJK_tmp_dim
8297   {
8298     \tex_unkern:D
8299     \__xeCJK_ulem_right_skip_glue:
8300     \tex_kern:D \l__xeCJK_tmp_dim
8301   }
8302 }
8303 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_glue:
8304 {
8305   \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8306   {
8307     \tex_unskip:D
8308     \skip_set:Nn \l__xeCJK_tmp_skip
8309       { \tex_lastskip:D - 2 \UL@pixel }
8310     \tex_unskip:D
8311     \dim_compare:nNnT { \l__xeCJK_tmp_skip } > { \c_zero_dim }
8312       { \UL@leadtype \skip_horizontal:N \l__xeCJK_tmp_skip }
8313     \skip_horizontal:N \UL@pixel
8314   }
8315 }
8316 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_penalty:
8317 {
8318   \int_set_eq:NN \l__xeCJK_tmp_int \tex_lastpenalty:D
8319   \tex_unpenalty:D
8320   \__xeCJK_if_last_hlist:T
8321     { \__xeCJK_ulem_right_skip_hbox: }
8322   \tex_penalty:D \l__xeCJK_tmp_int
8323 }

```

__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。

```

8324 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip:
8325 {
8326   \seq_gpop_right:NN \g__xeCJK_ulem_periodic_later_seq \l__xeCJK_tmp_tl
8327   \tl_if_eq:NnTF \l__xeCJK_tmp_tl { 1 }
8328     { \__xeCJK_ulem_periodic_right_skip_aux: }
8329     { \prg_do_nothing: }
8330 }
8331 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_aux:
8332 {
8333   \int_case:nn { \tex_lastnodetype:D }
8334     {
8335       { \c__xeCJK_hlist_node } { \__xeCJK_ulem_periodic_right_skip_hbox: }
8336       { \c__xeCJK_glue_node } { \__xeCJK_ulem_periodic_right_skip_glue: }
8337       { \c__xeCJK_penalty_node } { \__xeCJK_ulem_periodic_right_skip_penalty: }
8338     }

```

```

8339 }
8340 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_hbox:
8341 {
8342   \box_set_to_last:N \l__xeCJK_tmp_box
8343   \__xeCJK_if_last_kern:TF
8344     { \__xeCJK_ulem_periodic_right_skip_kern: }
8345     { \__xeCJK_ulem_periodic_right_skip_glue: }
8346   \box_use_drop:N \l__xeCJK_tmp_box
8347 }
8348 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_kern:
8349 {
8350   \dim_set:Nn \l__xeCJK_tmp_dim { - \box_wd:N \l__xeCJK_tmp_box }
8351   \dim_compare:nNnT \tex_lastkern:D = \l__xeCJK_tmp_dim
8352     {
8353       \tex_unkern:D
8354       \__xeCJK_ulem_periodic_right_skip_glue:
8355       \tex_kern:D \l__xeCJK_tmp_dim
8356     }
8357 }
8358 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_glue:
8359 {
8360   \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8361   {
8362     \tex_unskip:D
8363     \skip_set:Nn \l__xeCJK_tmp_skip { \tex_lastskip:D - 2 \UL@pixel }
8364     \tex_unskip:D
8365     \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
8366     {
8367       \tex_unskip:D
8368       \bool_if:NTF \l__xeCJK_ulem_subtract_bool
8369         {
8370           \__xeCJK_ulem_periodic_clipped_leaders:nnn
8371             { \l__xeCJK_tmp_skip } { - \UL@pixel } { \UL@pixel }
8372         }
8373         {
8374           \__xeCJK_ulem_periodic_clipped_leaders:nnn
8375             { \l__xeCJK_tmp_skip } { - \UL@pixel } { - \UL@pixel }
8376         }
8377     }
8378   }
8379 }
8380 \cs_new_protected:Npn \__xeCJK_ulem_periodic_right_skip_penalty:
8381 {
8382   \int_set_eq:NN \l__xeCJK_tmp_int \tex_lastpenalty:D
8383   \tex_unpenalty:D
8384   \__xeCJK_if_last_hlist:T
8385     { \__xeCJK_ulem_periodic_right_skip_hbox: }
8386   \tex_penalty:D \l__xeCJK_tmp_int
8387 }

```

__xeCJK_ulem_hidden_box: 只画线, 不输出盒子。

```

8388 \cs_new_protected:Npn \__xeCJK_ulem_hidden_box:
8389 {
8390   \tl_if_empty:NF \UL@start
8391   {
8392     \box_set_ht:Nn \l__xeCJK_hidden_box { \box_ht:N \UL@box }
8393     \box_set_dp:Nn \l__xeCJK_hidden_box { \box_dp:N \UL@box }
8394     \box_use:N \l__xeCJK_hidden_box
8395     \xeCJK_no_break:
8396     \xeCJK_ulem_hskip:n { \box_wd:N \UL@box }
8397     \box_use:N \l__xeCJK_hidden_box
8398   }
8399 }
8400 \box_new:N \l__xeCJK_hidden_box
8401 \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:
8402 \cs_new_protected:Npn \_xeCJK_ulem_skip_punct_begin:
8403 {
8404   \cs_set_eq:NN \UL@putbox \_xeCJK_ulem_skip_putbox:
8405   \cs_set_eq:NN \xeCJK_ulem_hskip:n \skip_horizontal:n
8406 }
8407 \cs_new_protected:Npn \_xeCJK_ulem_skip_punct_end:
8408 {
8409   \cs_set_eq:NN \UL@putbox \_xeCJK_ulem_putbox:
8410   \cs_set_eq:NN \xeCJK_ulem_hskip:n \_xeCJK_ulem_hskip_aux:n
8411 }
8412 \cs_new_eq:NN \_xeCJK_ulem_putbox: \UL@putbox
8413 \cs_new_protected:Npn \_xeCJK_ulem_skip_putbox:
8414 {
8415   \tl_if_empty:NF \UL@start
8416   { \box_use_drop:N \UL@box }
8417 }
```

_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。

```
8418 \cs_new_protected:Npn \_xeCJK_ulem_initial:
8419 {
8420   \_xeCJK_ulem_swap_cs:NN
8421   \xeCJK_FullLeft_and_Default: \_xeCJK_ulem_FullLeft_and_Default:
8422   \xeCJK_FullLeft_and_CJK: \_xeCJK_ulem_FullLeft_and_CJK:
8423   \xeCJK_FullLeft_and_Boundary: \_xeCJK_ulem_FullLeft_and_Boundary:
8424   \xeCJK_FullRight_and_Default: \_xeCJK_ulem_FullRight_and_Default:
8425   \xeCJK_FullRight_and_CJK: \_xeCJK_ulem_FullRight_and_CJK:
8426   \xeCJK_FullRight_and_CJStarter: \_xeCJK_ulem_FullRight_and_CJStarter:
8427   \xeCJK_FullRight_and_Boundary: \_xeCJK_ulem_FullRight_and_Boundary:
8428   \xeCJK_CJK_and_CJK:N \_xeCJK_ulem_CJK_and_CJK:N
8429   \xeCJK_CJK_and_Boundary:w \_xeCJK_ulem_CJK_and_Boundary:w
8430   \xeCJK@fix@penalty \_xeCJK_ulem_fix_penalty:
8431   \_xeCJK_punct_breakable_kern:n \_xeCJK_ulem_punct_breakable_kern:n
8432   \_xeCJK_Default_and_FullLeft_glue:N \_xeCJK_ulem_Default_and_FullLeft_glue:N
8433   \_xeCJK_Default_and_FullRight_glue:N \_xeCJK_ulem_Default_and_FullRight_glue:N
8434   \_xeCJK_CJK_and_FullLeft_glue:N \_xeCJK_ulem_CJK_and_FullLeft_glue:N
8435   \_xeCJK_CJK_and_FullRight_glue:N \_xeCJK_ulem_CJK_and_FullRight_glue:N
8436   \_xeCJK_Boundary_and_FullLeft_glue:N \_xeCJK_ulem_Boundary_and_FullLeft_glue:N
8437   \q_recursion_tail \q_nil \q_recursion_stop
8438   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
8439   {
8440     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
8441     {
8442       \str_if_eq:nnTF {##1} {####1}
8443       {
8444         \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
8445         { \_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
8446         \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
8447         { \_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
8448       }
8449       {
8450         \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
8451         { \_xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
8452       }
8453     }
8454   }
8455 }
8456 \cs_new_protected:Npn \_xeCJK_ulem_swap_cs:NN #1#2
8457 {
8458   \quark_if_recursion_tail_stop:N #1
8459   \xeCJK_swap_cs:NN #1#2
8460   \_xeCJK_ulem_swap_cs:NN
8461 }
```

`\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。

```
8462 \cs_new:Npn \xeCJK_if_ulem_patch:TF
8463 {
8464   \if_meaning:w \ \LA@space
8465   \exp_after:wN \use_ii:nn
8466   \else:
8467     \exp_after:wN \use_i:nn
8468   \fi:
8469 }
```

```
\_xeCJK_ulem_CJK_and_Boundary:w 8470 \cs_new_protected:Npn \_xeCJK_ulem_CJK_and_Boundary:w
8471 {
8472   \xeCJK_if_ulem_patch:TF
8473   {
8474     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
8475     { \xeCJK_class_group_end: \CJKeckglue }
8476     {
8477       \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
8478       { \_xeCJK_ulem_peek_math:w }
8479       { \_xeCJK_ulem_group_end:n { CJK } }
8480     }
8481   }
8482   { \_xeCJK_ulem_CJK_and_Boundary:w }
8483 }
8484 \cs_new_protected:Npn \_xeCJK_ulem_group_end:n #1
8485 {
8486   \xeCJK_class_group_end: \UL@stop
8487   \UL@start { \xeCJK_make_node:n {#1} }
8488   \xeCJK_make_group_tag:
8489   \_xeCJK_boundary_enable_source_space:n {#1}
8490 }
```

`\_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。

```
8491 \cs_new_protected:Npn \_xeCJK_ulem_peek_math:w
8492 {
8493   \cs_set_eq:NN \_xeCJK_ulem_start:w \_xeCJK_ulem_exp_stop:w
8494   \exp_after:wN \peek_after:Nw
8495   \exp_after:wN \_xeCJK_ulem_peek_math_branches:w
8496   \exp:w \exp_end_continue_f:w
8497 }
8498 \cs_new_protected:Npn \_xeCJK_ulem_peek_math_branches:w
8499 {
8500   \token_if_math_toggle:NTF \l_peek_token
8501   { \xeCJK_class_group_end: \CJKeckglue }
8502   { \_xeCJK_ulem_group_end:n { CJK-space } }
8503 }
8504 \cs_new_protected:Npn \_xeCJK_ulem_exp_stop:w
8505 {
8506   \cs_if_eq:NNTF \UL@start \@empty
8507   { \exp_after:wN \exp_stop_f: }
8508   { \exp_after:wN \UL@start }
8509 }
```

```
\_xeCJK_ulem_fix_penalty: 8510 \cs_new_protected:Npn \_xeCJK_ulem_fix_penalty:
8511 {
8512   \xeCJK_if_ulem_patch:TF
8513   { \fix@penalty }
8514   { \_xeCJK_ulem_fix_penalty: }
8515 }
```

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

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:
\_xeCJK_ulem_restore_font_state:
\g__xeCJK_ulem_saved_series_tl
\g__xeCJK_ulem_saved_shape_tl
\g__xeCJK_ulem_saved_CJK_family_tl
\g__xeCJK_ulem_saved_CJK_fam_tl
8516 \tl_new:N \g__xeCJK_ulem_saved_series_tl
8517 \tl_new:N \g__xeCJK_ulem_saved_shape_tl
8518 \tl_new:N \g__xeCJK_ulem_saved_CJK_family_tl
8519 \tl_new:N \g__xeCJK_ulem_saved_CJK_fam_tl
8520 \cs_new_protected:Npn \_xeCJK_ulem_save_font_state:
8521 {
8522   \tl_gset:Ne \g__xeCJK_ulem_saved_series_tl { \f@series }
8523   \tl_gset:Ne \g__xeCJK_ulem_saved_shape_tl { \f@shape }
8524   \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_family_tl \l_xeCJK_family_tl
8525   \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_fam_tl \CJK@family
8526 }
8527 \cs_new_protected:Npn \_xeCJK_ulem_restore_font_state:
8528 {
8529   \tl_set:Ne \f@series { \g__xeCJK_ulem_saved_series_tl }
8530   \tl_set:Ne \f@shape { \g__xeCJK_ulem_saved_shape_tl }
8531   \tl_set_eq:NN \l_xeCJK_family_tl \g__xeCJK_ulem_saved_CJK_family_tl
8532   \tl_set_eq:NN \CJK@family \g__xeCJK_ulem_saved_CJK_fam_tl
8533 }

\_xeCJK_ulem_CJK_and_CJK:N
8534 \cs_new_protected:Npn \_xeCJK_ulem_CJK_and_CJK:N
8535 {
8536   \xeCJK_if_ulem_patch:TF
8537   {
8538     \_xeCJK_ulem_save_font_state:
8539     \xeCJK_class_group_end:
8540     \UL@stop \_xeCJK_ulem_ccglue: \UL@start
8541     \_xeCJK_ulem_class_group_begin:
8542     \_xeCJK_ulem_restore_font_state:
8543     \xeCJK_select_font:
8544     \xeCJK_fallback_symbol:NN
8545     \CJKsymbol
8546   }
8547   { \_xeCJK_ulem_CJK_and_CJK:N }
8548 }

\_xeCJK_ulem_class_group_begin:
8549 \cs_new_protected:Npn \_xeCJK_ulem_class_group_begin:
8550 {
8551   \xeCJK_class_group_begin:
8552   \xeCJK_clear_Boundary_and_CJK_toks:
8553 }

\_xeCJK_ulem_between_CJK_blocks:nnN
8554 \cs_new_protected:Npn \_xeCJK_ulem_between_CJK_blocks:nnN #1#2
8555 {
8556   \xeCJK_if_ulem_patch:TF
8557   {
8558     \_xeCJK_ulem_save_font_state:
8559     \xeCJK_class_group_end:
8560     \UL@stop \_xeCJK_ulem_ccglue: \UL@start
8561     \xeCJK_class_group_begin:
8562     \xeCJK_clear_Boundary_and_CJK_toks:
8563     \_xeCJK_ulem_restore_font_state:
8564     \_xeCJK_switch_font:nn {#1} {#2}
8565     \xeCJK_fallback_symbol:NN
8566     \CJKsymbol
8567   }
8568   {
8569     \skip_horizontal:N \l__xeCJK_ccglue_skip
8570     \_xeCJK_switch_font:nn {#1} {#2}
8571     \xeCJK_fallback_symbol:NN
8572     \CJKsymbol
8573   }
8574 }

\_xeCJK_ulem_Default_and_FullLeft_glue:N
8575 \cs_new_protected:Npn \_xeCJK_ulem_Default_and_FullLeft_glue:N #1
8576 {

```

```

8577 \xeCJK_if_ulem_patch:TF
8578 {
8579     \UL@stop
8580     \__xeCJK_ulem_skip_punct_begin:
8581     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8582     \UL@start
8583 }
8584 { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
8585 }

\__xeCJK_ulem_Boundary_and_FullLeft_glue:N 8586 \cs_new_protected:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
8587 {
8588     \xeCJK_if_ulem_patch:TF
8589     {
8590         \UL@stop
8591         \__xeCJK_ulem_skip_punct_begin:
8592         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8593         \UL@start
8594     }
8595     { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
8596 }

\__xeCJK_ulem_CJK_and_FullLeft_glue:N 8597 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
8598 {
8599     \xeCJK_if_ulem_patch:TF
8600     {
8601         \xeCJK_class_group_end:
8602         \UL@stop
8603         \__xeCJK_ulem_skip_punct_begin:
8604         \__xeCJK_ulem_punct_ccglue:
8605         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
8606         \UL@start
8607         \__xeCJK_ulem_class_group_begin:
8608         \xeCJK_select_punct_font:
8609     }
8610     { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
8611 }

\__xeCJK_ulem_Default_and_FullRight_glue:N 8612 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
8613 {
8614     \xeCJK_if_ulem_patch:TF
8615     {
8616         \UL@stop
8617         \__xeCJK_ulem_skip_punct_begin:
8618         \__xeCJK_punct_if_long:NTF #1
8619         { \xeCJK_allow_break: }
8620         { \xeCJK_no_break: }
8621         \__xeCJK_punct_if_middle:NT #1
8622         {
8623             \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
8624             \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
8625         }
8626         \UL@start
8627     }
8628     { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
8629 }

\__xeCJK_ulem_CJK_and_FullRight_glue:N 8630 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
8631 {
8632     \xeCJK_if_ulem_patch:TF
8633     {
8634         \xeCJK_class_group_end:
8635         \UL@stop
8636         \__xeCJK_ulem_skip_punct_begin:
8637         \__xeCJK_punct_if_long:NTF #1
8638         { \xeCJK_allow_break: }
8639         { \xeCJK_no_break: }
8640         \__xeCJK_punct_if_middle:NT #1

```

```

8641         {
8642             \_xeCJK_ulem_punct_ccglue:
8643             \_xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
8644             \_xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
8645         }
8646         \UL@start
8647         \_xeCJK_ulem_class_group_begin:
8648         \xeCJK_select_punct_font:
8649     }
8650     { \_xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
8651 }

\_xeCJK_ulem_FullLeft_and_Default: 8652 \cs_new_protected:Npn \_xeCJK_ulem_FullLeft_and_Default:
8653 {
8654     \xeCJK_if_ulem_patch:TF
8655     {
8656         \_xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
8657         {
8658             \xeCJK_get_punct_bounds:Np \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8659             \_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8660             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
8661             \_xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8662         }
8663         { \xeCJK_class_group_end: \UL@stop }
8664         \_xeCJK_ulem_skip_punct_end:
8665         \xeCJK_no_break:
8666         \UL@start
8667     }
8668     { \_xeCJK_ulem_FullLeft_and_Default: }
8669 }

\_xeCJK_ulem_FullLeft_and_Boundary: 8670 \cs_new_protected:Npn \_xeCJK_ulem_FullLeft_and_Boundary:
8671 {
8672     \xeCJK_if_ulem_patch:TF
8673     {
8674         \_xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
8675         {
8676             \xeCJK_get_punct_bounds:Np \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8677             \_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8678             \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
8679             \_xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
8680         }
8681         { \xeCJK_class_group_end: \UL@stop }
8682         \_xeCJK_ulem_skip_punct_end:
8683         \xeCJK_no_break:
8684         \UL@start
8685         \tex_ignorespaces:D
8686     }
8687     { \_xeCJK_ulem_FullLeft_and_Boundary: }
8688 }

\_xeCJK_ulem_FullLeft_and_CJK: 8689 \cs_new_protected:Npn \_xeCJK_ulem_FullLeft_and_CJK:
8690 {
8691     \xeCJK_if_ulem_patch:TF
8692     {
8693         \xeCJK_FullLeft_and_Default:
8694         \_xeCJK_ulem_class_group_begin:
8695         \xeCJK_select_font:
8696     }
8697     { \_xeCJK_ulem_FullLeft_and_CJK: }
8698 }

\_xeCJK_ulem_FullRight_and_Default: 8699 \cs_new_protected:Npn \_xeCJK_ulem_FullRight_and_Default:
8700 {
8701     \xeCJK_if_ulem_patch:TF
8702     {
8703         \_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8704         \xeCJK_class_group_end:

```

```

8705         \UL@stop
8706         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8707         \__xeCJK_ulem_skip_punct_end:
8708         \UL@start
8709     }
8710     { \__xeCJK_ulem_FullRight_and_Default: }
8711 }

```

```

\__xeCJK_ulem_FullRight_and_Boundary: 8712 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Boundary:
8713 {
8714     \xeCJK_if_ulem_patch:TF
8715     {
8716         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8717         \xeCJK_class_group_end:
8718         \UL@stop
8719         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8720         \__xeCJK_ulem_skip_punct_end:
8721         \UL@start
8722         \tex_ignorespaces:D
8723     }
8724     { \__xeCJK_ulem_FullRight_and_Boundary: }
8725 }

```

```

\__xeCJK_ulem_FullRight_and_CJK: 8726 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJK:
8727 {
8728     \xeCJK_if_ulem_patch:TF
8729     {
8730         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8731         \xeCJK_class_group_end:
8732         \UL@stop
8733         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8734         \__xeCJK_ulem_punct_ccglue:
8735         \__xeCJK_ulem_skip_punct_end:
8736         \UL@start
8737         \__xeCJK_ulem_class_group_begin:
8738         \xeCJK_select_font:
8739     }
8740     { \__xeCJK_ulem_FullRight_and_CJK: }
8741 }

```

__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。

```

8742 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJStarter:
8743 {
8744     \xeCJK_if_ulem_patch:TF
8745     {
8746         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8747         \xeCJK_class_group_end:
8748         \UL@stop
8749         \xeCJK_no_break:
8750         \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
8751         \__xeCJK_ulem_punct_ccglue:
8752         \__xeCJK_ulem_skip_punct_end:
8753         \UL@start
8754         \__xeCJK_ulem_class_group_begin:
8755         \xeCJK_select_font:
8756     }
8757     { \__xeCJK_ulem_FullRight_and_CJStarter: }
8758 }

```

```

\__xeCJK_ulem_punct_hskip:n 8759 \cs_new_protected:Npn \__xeCJK_ulem_punct_hskip:n
8760 {
8761     \xeCJK_if_ulem_patch:TF
8762     { \xeCJK_ulem_hskip:n }
8763     { \__xeCJK_ulem_punct_hskip:n }
8764 }

```

```

\__xeCJK_ulem_punct_breakable_kern:n 8765 \cs_new_protected:Npn \__xeCJK_ulem_punct_breakable_kern:n #1
8766 {
8767   \xeCJK_if_ulem_patch:TF
8768   {
8769     \xeCJK_class_group_end:
8770     \UL@stop \xeCJK_ulem_hskip:n {#1} \UL@start
8771     \__xeCJK_ulem_class_group_begin:
8772     \xeCJK_select_punct_font:
8773   }
8774   { \__xeCJK_ulem_punct_breakable_kern:n {#1} }
8775 }

```

```

\__xeCJK_ulem_glue:n 在下划线状态下的分别代替 \CJKglue 等。
\__xeCJK_ulem_ccglue:n
\__xeCJK_ulem_punct_ccglue:n
8776 \cs_new_protected:Npn \__xeCJK_ulem_glue:n #1
8777 {
8778   \xeCJK_if_ulem_patch:TF
8779   {
8780     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
8781     { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
8782     {
8783       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
8784       { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
8785       { \__xeCJK_ulem_defer_glue:n {#1} }
8786     }
8787   }
8788   { \skip_horizontal:n {#1} }
8789 }

```

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 `\CJKunderline{中}\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。

```

8790 \cs_gset_protected:Npn \__xeCJK_use_ecglue_skip:
8791 {
8792   \bool_lazy_and:nnTF
8793   { \l__xeCJK_ulem_stream_started_bool }
8794   { \cs_if_eq_p:NN \UL@start \@empty }
8795   { \__xeCJK_ulem_glue:n { \l__xeCJK_ecglue_skip } }
8796   { \skip_horizontal:N \l__xeCJK_ecglue_skip }
8797 }

```

```

\__xeCJK_ulem_defer_glue:n 这两个函数处理 #1067: 装饰正文里用花括号括住一段内容时, 该处间距的可收缩量原先会失
\__xeCJK_ulem_flush_pending_shrink:n 效。

```

成因不在 ulem 的切词——实测两种写法切出的词相同, 只差一对花括号。花括号是在词内容交给 \UL@start 之后、在片段盒内部才展开成分组的, 于是边界检测在盒内、且在用户分组内触发; _xeCJK_ulem_glue:n 的 group tag 守卫比对保存的 T1L4 与当前的 T1L5 不相等, 走 else 分支。该分支原先是裸 \skip_horizontal:n, 间距连收缩量一起固化在定宽片段盒内, 外层取不到。

不能改成「照样搬运」: 实测强制走 \UL@stop 之后, 搬出来的 glue 落进 \cleaders 内部、仍在盒内——在用户分组内 \UL@stop 关不掉正确的层级(ulem 开盒时开了两层, 用户花括号插在中间), \UL@start 重开的盒子又把它包了回去。而且绕过守卫会让分组内的字体设置丢失(fntef-font01 即此, 字体从楷体退回宋体)。也就是说「守卫是直接原因」与「绕过守卫无效」两件事同时成立, 修法必须绕开这个结构, 而不是拆掉守卫。

因此改为把一处间距拆成两半输出:

- 盒内放一个不可伸缩的 \kern, 占住自然宽度, 使排版位置与盒子总宽都不变;
- 把伸缩量记进全局的 \g__xeCJK_ulem_pending_shrink_skip, 等到 _xeCJK_ulem_loop:nw 的词尾搬运处——那里已经在片段盒外、用户分组外——再补一个零宽而带伸缩的 glue。

短路条件是整个 skip 全零(\skip_if_eq:nnTF 比对 \c_zero_skip), 不是「自然宽度为零」。零自然宽度但带弹性的间距仍然要记账搬运——用户可以把间距设成 CJKecglue={\hskip 0pt minus 此时自然宽度是零而收缩量非零, 若按自然宽度短路, 这份收缩量就留在定宽片段盒里出不来(实测压窄 1.2pt: 直接输入与无编组装饰都是 18, 而分组形态是 1000000)。

这条短路只用来跳过真正什么都不做的情形, 避免为零 glue 白记一笔账。

于是 \CJKglue (0pt plus 0.96, 只有伸长量) 也走记账通道, 它的伸长量随之回到外层列表。这是有意的: 分组内的伸长量因此同样能参与断行, 与不带装饰时的行为更接近。实测 \CJKunderline{文字{\sffamily 文字}} 拉伸 3pt 的 badness 从 3049 降到 381, 向无装饰形态的 113 靠拢, 而盒子宽度不变(40.0pt)。fntef-font01 的基线因此记录 \kern 而不再是原处的 \glue。

取弹性部分用整个 skip 减去自然宽度(\skip_eval:n 里的 #1 - \dim_eval:n {#1}), 不要分别取 \gluestretch 与 \glueshrink 再用 plus / minus 重新拼: 那两个原语只返回数值, 阶数在 \gluestretchorder / \glueshrinkorder 里, 分开取会把 1fil 降成普通阶(实测 3.33pt plus 1fil minus 1.11pt 用减法得到 0.0pt plus 1.0fil minus 1.11pt, 阶数保留; 而 \gluestretch 只给出 1.0pt)。就目前的调用路径而言这两种写法结果相同——装饰状态下 \l__xeCJK_ecglue_skip 由 \xeCJK_glue_to_skip:nN 现算, 传进来的自然宽度已是零、会走上上面那条短路——但减法写法不依赖这个前提, 日后间距模板换成带无限阶伸缩的形式也不会静默降阶。

记账是全局量, 因此在两处清零: _xeCJK_ulem_stream_begin: 进入装饰时清一次, _xeCJK_ulem_end: 结束时再清一次。两处都要, 不是重复——结束处那次覆盖正常路径, 而行内公式里的装饰命令经 \UL@onmath / \UL@onin 结束、不走 _xeCJK_ulem_end:, 那条路径上的残留只能靠下一次装饰开始时清掉。实测目前四条路径(公式内带编组、公式内 \mbox、\mbox 内装饰、装饰嵌套)的残留都是 0pt, 因为记账在词尾都已兑现; 开始处这道清零是防御性的, 用来隔断将来改动公式路径时可能出现的串味。

补出的 glue 必须放在 \UL@stop 之后、\UL@leaders 之前。 \UL@leaders 的结构是「-\UL@pixel、加宽两个 pixel 的 leaders、再 -\UL@pixel」, 末尾那个负值是段与段之间的重叠补偿, 要靠紧随其后的内容抵掉。如果把补出的 glue 放在 \UL@leaders 之后, 编组位于装饰参数结尾时它就会把这个负值推到更后面, 使其失去正向配对, 装饰盒因此比实际内容窄一个 pixel(实测 \CJKunderline{中{x}} 得 17.61002pt, 而 \hbox{中{x}} 是 18.61pt)。放在 \UL@leaders 之前, glue 落在两段装饰之间, 两种位置都正确。fntef-shrink01 的 TEST 11 同时固定「编组在中间」与「编组在结尾」两种位置, 正是因为前者察觉不到这个回归。

8798 \skip_new:N \g__xeCJK_ulem_pending_shrink_skip

8799 \cs_new_protected:Npn _xeCJK_ulem_defer_glue:n #1

```

8800 {
8801   \skip_if_eq:nnTF { #1 } { \c_zero_skip }
8802   { \skip_horizontal:n {#1} }
8803   {
8804     \skip_gadd:Nn \g__xeCJK_ulem_pending_shrink_skip
8805     { \skip_eval:n { #1 - \dim_eval:n {#1} } }
8806     \tex_kern:D \dim_eval:n {#1} \scan_stop:
8807   }
8808 }
8809 \cs_new_protected:Npn \__xeCJK_ulem_flush_pending_shrink:
8810 {
8811   \skip_if_eq:nnF { \g__xeCJK_ulem_pending_shrink_skip } { \c_zero_skip }
8812   {
8813     \skip_horizontal:N \g__xeCJK_ulem_pending_shrink_skip
8814     \skip_gzero:N \g__xeCJK_ulem_pending_shrink_skip
8815   }
8816 }
8817 \cs_new_protected:Npn \xeCJK_make_group_tag:
8818 { \tl_set:Nn \l__xeCJK_group_tag_tl { \c__xeCJK_group_tag_tl } }
8819 \tl_new:N \l__xeCJK_group_tag_tl
8820 \tl_const:Nn \c__xeCJK_group_tag_tl
8821 {
8822   T \int_use:N \tex_currentgroupstype:D
8823   L \int_use:N \tex_currentgrouplevel:D
8824 }
8825 \cs_new_protected:Npn \__xeCJK_ulem_ccglue:
8826 { { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders } }
8827 \cs_new_protected:Npn \__xeCJK_ulem_punct_ccglue:
8828 { \__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。`\ULon/\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。

```

8829 \cs_new_protected:Npn \__xeCJK_boundary_use_ulem_glue:n #1
8830 {
8831   \xeCJK_if_ulem_patch:TF
8832   {
8833     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
8834     { \UL@stop \skip_horizontal:n {#1} \UL@start }
8835     {
8836       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
8837       { \UL@stop \skip_horizontal:n {#1} \UL@start }
8838       { \skip_horizontal:n {#1} }
8839     }
8840   }
8841   { \skip_horizontal:n {#1} }
8842 }
8843 \cs_new_protected:Npn \__xeCJK_ulem_stream_begin:
8844 {
8845   \bool_if:NF \l__xeCJK_ulem_stream_started_bool
8846   {
8847     \bool_set_true:N \l__xeCJK_ulem_stream_started_bool
8848     \skip_gzero:N \g__xeCJK_ulem_pending_shrink_skip
8849     \__xeCJK_boundary_inline_stream_begin:nn { auto } { stream-ulem }
8850   }

```

```

8851 }
8852 \cs_new_protected:Npn \xeCJK_ulem_group_begin:
8853 {
8854   \mode_leave_vertical:
8855   \__xeCJK_ulem_stream_begin:
8856   \c_group_begin_token
8857 }
8858 \cs_new_protected:Npn \xeCJK_ulem_group_end:
8859 { \c_group_end_token }
8860 \cs_new_protected:Npn \xeCJK_ulem_on:n
8861 { \ULon }
8862 \cs_new_eq:NN \__xeCJK_orig_ulem_on: \ULon
8863 \cs_set_protected:Npn \ULon
8864 {
8865   \__xeCJK_ulem_stream_begin:
8866   \__xeCJK_orig_ulem_on:
8867 }
8868 \bool_new:N \l__xeCJK_ulem_stream_started_bool

```

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, 实际末节点检查仍会失败。

无论走哪条路径, 正文都必须以字面记号进入 \UL@on 的替换文本, 不能再经过一层宏参数间接展开: ulem 自己扫描正文并按源码空格切分片段, 正文若作为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 \CJKecglue 会落在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, 整段可收缩量因此按西文词数减少(#1026)。

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完整的 ulem 参数, 再用 \exp_args:NV 把它一次展开到参数位置, 使记号与直接书写 #1 等价; 不能改用 \tl_use:N, 那会让正文晚一层展开而重新触发上述问题。 \UL@on 需要两端的 \xeCJK_ulem_left: / \xeCJK_ulem_right:, \UL@onin 由 ulem 自己的 \UL@onin 负责这两端, 因此拼装时把它们作为前后缀参数传入, 不写死在 body 里。

```

8869 \tl_new:N \l__xeCJK_ulem_body_tl
8870 \cs_new_protected:Npn \__xeCJK_boundary_ulem_math_tail_space:nnn #1#2#3
8871 {
8872   \tl_set:Nn \l__xeCJK_ulem_body_tl {#2}
8873   \tl_put_right:Ne \l__xeCJK_ulem_body_tl
8874     { \tl_trim_right_spaces:n {#1} }
8875   \tl_put_right:Nn \l__xeCJK_ulem_body_tl
8876     {
8877       \__xeCJK_boundary_if_last_math_node:T
8878       { \__xeCJK_boundary_math_end_confirm_space_visible: }
8879       \c_space_tl
8880     }
8881   \tl_put_right:Nn \l__xeCJK_ulem_body_tl {#3}
8882 }
8883 \prg_new_protected_conditional:Npnn \__xeCJK_boundary_if_ulem_math_reorder:n #1
8884 { T , F , TF }
8885 {
8886   \__xeCJK_boundary_if_math_tail:nTF {#1}
8887   {
8888     \__xeCJK_boundary_if_math_tail_space:nTF {#1}
8889     { \prg_return_true: }
8890     { \prg_return_false: }
8891   }
8892   { \prg_return_false: }
8893 }
8894 \cs_new_eq:NN \__xeCJK_ulem_on:n \UL@on
8895 \cs_set_protected:Npn \UL@on #1
8896 {
8897   \__xeCJK_boundary_math_begin:n {#1}
8898   \__xeCJK_boundary_if_ulem_math_reorder:nTF {#1}

```

```

8899     {
8900       \__xeCJK_boundary_ulem_math_tail_space:nnn {#1}
8901       { \xeCJK_ulem_left: } { \xeCJK_ulem_right: }
8902       \exp_args:NV \__xeCJK_ulem_on:n \l__xeCJK_ulem_body_tl
8903     }
8904     {
8905       \__xeCJK_ulem_on:n
8906       {
8907         \xeCJK_ulem_left: #1
8908         \__xeCJK_boundary_math_end:n {#1}
8909         \xeCJK_ulem_right:
8910       }
8911     }
8912   }
8913   \cs_new_eq:NN \__xeCJK_orig_ulem_onin:n \UL@onin
8914   \cs_set_protected:Npn \UL@onin #1
8915   {
8916     \__xeCJK_boundary_math_begin:n {#1}
8917     \__xeCJK_boundary_if_ulem_math_reorder:nTF {#1}
8918     {
8919       \__xeCJK_boundary_ulem_math_tail_space:nnn {#1} { } { }
8920       \exp_args:NV \__xeCJK_orig_ulem_onin:n \l__xeCJK_ulem_body_tl
8921     }
8922     { \__xeCJK_orig_ulem_onin:n { #1 \__xeCJK_boundary_math_end:n {#1} } }
8923   }

```

\xeCJKfntefon 扩展 \ULon 的参数。

```

8924 \NewDocumentCommand \xeCJKfntefon { s t- s o }
8925 {
8926   \mode_leave_vertical:
8927   \xeCJK_ulem_boot:NNNn #1#2#3 {#4}
8928   \xeCJK_ulem_on:n
8929 }

```

\CJKKunderline

```

8930 \NewDocumentCommand \CJKKunderline { s t- s o }
8931 {
8932   \xeCJK_ulem_group_begin:
8933   \xeCJK_fntef_boot:nnNNn { underline } { uline } #1#2#3 {#4}
8934   \cs_set_eq:NN \ULleaders \tex_cleaders:D
8935   \xeCJK_fntef_initial:nnn
8936   { \l__xeCJK_uline_depth_tl }
8937   { \l__xeCJK_uline_sep_tl }
8938   {
8939     \l__xeCJK_uline_format_tl
8940     \tex_vrule:D
8941     height \dim_eval:n { \l__xeCJK_uline_thickness_tl }
8942     depth \c_zero_dim
8943     width .2em
8944   }
8945   \xeCJK_ulem_on:n
8946 }
8947 \NewDocumentCommand \varCJKKunderline { }
8948 { \CJKKunderline - }

```

`\__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。

```

8949 \cs_new_protected:Npn \__xeCJK_fntef_wave_symbol:
8950 {
8951   \dim_set:Nn \l__xeCJK_fntef_pattern_dim { 1 em / 4 }
8952   \hbox_to_wd:nn { \l__xeCJK_fntef_pattern_dim }
8953   {
8954     \hss
8955     \draw_begin:
8956       \draw_set_linewidth:n { .04 em }
8957       \draw_set_cap_but:
8958       \draw_path_moveto:n { 0 pt , .055 em }
8959       \draw_path_curveto:nnn
8960         { \l__xeCJK_fntef_pattern_dim * 3 / 16 , .055 em }
8961         { \l__xeCJK_fntef_pattern_dim * 5 / 16 , -.055 em }
8962         { \l__xeCJK_fntef_pattern_dim / 2 , -.055 em }
8963       \draw_path_curveto:nnn
8964         { \l__xeCJK_fntef_pattern_dim * 11 / 16 , -.055 em }
8965         { \l__xeCJK_fntef_pattern_dim * 13 / 16 , .055 em }
8966         { \l__xeCJK_fntef_pattern_dim , .055 em }
8967       \draw_path_use_clear:n { draw }
8968     \draw_end:
8969     \hss
8970   }
8971 }
8972 \cs_new_protected:Npn \__xeCJK_fntef_xout_symbol:
8973 {
8974   \dim_set:Nn \l__xeCJK_fntef_pattern_dim { 1 em / 4 }
8975   \hbox_to_wd:nn { \l__xeCJK_fntef_pattern_dim }
8976   {
8977     \hss
8978     \draw_begin:
8979       \draw_set_linewidth:n { .04 em }
8980       \draw_set_cap_but:
8981       \draw_path_moveto:n { 0 pt , -.445 em }
8982       \draw_path_lineto:n
8983         { \l__xeCJK_fntef_pattern_dim , .445 em }
8984       \draw_path_use_clear:n { draw }
8985     \draw_end:
8986     \hss
8987   }
8988 }
8989 \dim_new:N \l__xeCJK_fntef_pattern_dim

\CJKunderwave 8990 \NewDocumentCommand \CJKunderwave { s t- s o }
8991 {
8992   \xeCJK_ulem_group_begin:
8993     \xeCJK_fntef_boot:nnNNNn { underwave } { uwave } #1#2#3 {#4}
8994     \cs_set_eq:NN \ULleaders \tex_xleaders:D
8995     \tl_if_eq:NnT \l__xeCJK_uwave_symbol_tl { \__xeCJK_fntef_wave_symbol: }
8996     {
8997       \cs_set_eq:NN \ULleaders \tex_leaders:D
8998       \bool_set_true:N \l__xeCJK_ulem_periodic_bool
8999     }
9000     \xeCJK_fntef_initial:nnn
9001     { \l__xeCJK_uwave_depth_tl }
9002     { \l__xeCJK_uwave_sep_tl }
9003     { \l__xeCJK_uwave_format_tl \l__xeCJK_uwave_symbol_tl }
9004     \xeCJK_ulem_on:n
9005   }

\CJKunderdblline 9006 \NewDocumentCommand \CJKunderdblline { s t- s o }
9007 {
9008   \xeCJK_ulem_group_begin:

```

```

9009     \xeCJK_fntef_boot:nnNNNn { underdblline } { udbline } #1#2#3 {#4}
9010     \cs_set_eq:NN \ULleaders \tex_cleaders:D
9011     \xeCJK_fntef_initial:nnn
9012     { \l__xeCJK_udbline_depth_tl }
9013     { \l__xeCJK_udbline_sep_tl }
9014     {
9015         \l__xeCJK_udbline_format_tl
9016         \vbox_top:n
9017         {
9018             \tex_hrulerule:D
9019             height \dim_eval:n { \l__xeCJK_udbline_thickness_tl }
9020             depth \c_zero_dim
9021             width .2em
9022             \tex_kern:D \dim_eval:n { \l__xeCJK_udbline_gap_tl }
9023             \tex_hrulerule:D
9024             height \dim_eval:n { \l__xeCJK_udbline_thickness_tl }
9025             depth \c_zero_dim
9026             width .2em
9027         }
9028     }
9029     \xeCJK_ulem_on:n
9030 }

\CJKsout 9031 \NewDocumentCommand \CJKsout { s t- s o }
9032 {
9033     \xeCJK_ulem_group_begin:
9034     \xeCJK_fntef_boot:nnNNNn { sout } { sout } #1#2#3 {#4}
9035     \cs_set_eq:NN \ULleaders \tex_cleaders:D
9036     \xeCJK_fntef_initial:nn
9037     {
9038         \l__xeCJK_sout_format_tl
9039         \tex_vrulerule:D
9040         height \dim_eval:n { \l__xeCJK_sout_thickness_tl }
9041         depth \c_zero_dim
9042         width .2em
9043     }
9044     {
9045         \box_move_up:nn
9046         { \l__xeCJK_sout_height_tl - \box_ht:N \l__xeCJK_fntef_box / 2 }
9047         { \box_use:N \l__xeCJK_fntef_box }
9048     }
9049     \xeCJK_ulem_on:n
9050 }

\CJKxout 9051 \NewDocumentCommand \CJKxout { s t- s o }
9052 {
9053     \xeCJK_ulem_group_begin:
9054     \xeCJK_fntef_boot:nnNNNn { xout } { xout } #1#2#3 {#4}
9055     \cs_set_eq:NN \ULleaders \tex_leaders:D
9056     \xeCJK_fntef_initial:nn
9057     {
9058         \l__xeCJK_xout_format_tl
9059         \__xeCJK_fntef_xout_symbol:
9060     }
9061     {
9062         \box_move_down:nn
9063         { .09 em }
9064         { \box_use:N \l__xeCJK_fntef_box }
9065     }
9066     \bool_set_true:N \l__xeCJK_ulem_periodic_bool
9067     \xeCJK_ulem_on:n
9068 }

\CJKunderanyline 9069 \NewDocumentCommand \CJKunderanyline { s t- s o m m }
9070 {
9071     \xeCJK_ulem_group_begin:
9072     \xeCJK_ulem_boot:NNNn #1#2#3 {#4}
9073     \cs_set_eq:NN \ULleaders \tex_cleaders:D

```

```

9074 \xeCJK_fntef_initial:nn
9075 {#6}
9076 {
9077   \box_move_down:nn
9078   {#5}
9079   { \box_use:N \l__xeCJK_fntef_box }
9080 }
9081 \tl_if_empty:NF \l__xeCJK_ulem_boxdepth_tl
9082 { \box_set_dp:Nn \ULC@box { \l__xeCJK_ulem_boxdepth_tl } }
9083 \tl_if_empty:NF \l__xeCJK_ulem_sep_tl
9084 {
9085   \bool_set_true:N \l__xeCJK_fntef_bool
9086   \dim_set:Nn \l__xeCJK_fntef_dim
9087   { \l__xeCJK_ulem_sep_tl + \box_dp:N \ULC@box }
9088 }
9089 \xeCJK_ulem_on:n
9090 }

```

\xeCJK_fntef_boot:nnNNn 处理参数问题。

```

9091 \cs_new_protected:Npn \xeCJK_fntef_boot:nnNNn #1#2#3#4#5#6
9092 {
9093   \bool_lazy_or:nnT {#3} {#5}
9094   { \bool_set_false:c { l__xeCJK_#2_skip_bool } }
9095   \bool_if:NT #4
9096   { \bool_set_true:c { l__xeCJK_#2_subtract_bool } }
9097   \tl_if_novalue:NF {#6}
9098   { \keys_set:nn { xeCJK / options / #1 } {#6} }
9099   \bool_set_eq:Nc \l__xeCJK_ulem_skip_bool { l__xeCJK_#2_skip_bool }
9100   \bool_set_eq:Nc \l__xeCJK_ulem_hidden_bool { l__xeCJK_#2_hidden_bool }
9101   \bool_set_eq:Nc \l__xeCJK_ulem_subtract_bool { l__xeCJK_#2_subtract_bool }
9102   \tl_set_eq:Nc \l__xeCJK_ulem_text_format_tl { l__xeCJK_#2_text_format_tl }
9103 }
9104 \cs_new_protected:Npn \xeCJK_ulem_boot:NNNn #1#2#3#4
9105 {
9106   \bool_lazy_or:nnT {#1} {#3}
9107   { \bool_set_false:N \l__xeCJK_ulem_skip_bool }
9108   \bool_if:NT #2
9109   { \bool_set_true:N \l__xeCJK_ulem_subtract_bool }
9110   \tl_if_novalue:NF {#4}
9111   { \keys_set:nn { xeCJK / options / ulem } {#4} }
9112 }

```

\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。

```

9113 \cs_new_protected:Npn \xeCJK_fntef_initial:n
9114 {
9115   \bool_if:NTF \l__xeCJK_nest_bool
9116   { \__xeCJK_warning:n { fntef-nesting } }
9117   {
9118     \bool_set_true:N \l__xeCJK_nest_bool
9119     \__xeCJK_restore_shipout_fntef:
9120   }
9121   \xeCJK_fntef_sbox:n
9122 }
9123 \cs_new_protected:Npn \xeCJK_fntef_initial:nn #1
9124 {
9125   \xeCJK_fntef_initial:n {#1}
9126   \bool_if:NF \l__xeCJK_fntef_bool
9127   { \dim_zero:N \l__xeCJK_fntef_dim }
9128   \markoverwith
9129 }
9130 \cs_new_protected:Npn \xeCJK_fntef_initial:nnn #1#2#3
9131 {
9132   \xeCJK_fntef_initial:n {#3}
9133   \bool_if:NF \l__xeCJK_fntef_bool
9134   {
9135     \bool_set_true:N \l__xeCJK_fntef_bool

```

```

9136     \dim_set:Nn \l__xeCJK_fntef_dim {#1}
9137   }
9138   \markoverwith
9139   {
9140     \box_move_down:nn
9141     { \l__xeCJK_fntef_dim + \box_ht:N \l__xeCJK_fntef_box }
9142     { \box_use:N \l__xeCJK_fntef_box }
9143   }
9144   \dim_set:Nn \l__xeCJK_fntef_dim { #2 + \box_dp:N \ULC@box }
9145 }
9146 \box_new:N \l__xeCJK_fntef_box
9147 \cs_new_eq:NN \xeCJKfntefbox \l__xeCJK_fntef_box
9148 \bool_new:N \l__xeCJK_nest_bool
9149 \bool_new:N \l__xeCJK_fntef_bool
9150 \__xeCJK_msg_new:nn { fntef-nesting }
9151 { 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。

```
9152 \cs_new_eq:NN \l__xeCJK_fntef_dim \ULdepth
```

`\xeCJK_fntef_sbox:n` 与 `\hcoffin_set:Nn` 和 $\text{\LaTeX 2}_{\epsilon}$ 的 `\sbox` 功能类似,确保颜色的正确。虽然 `coffin` 可以更方便的操作盒子,但速度要慢一点。并且,我们的需求也比较简单,就不用它了。

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。

```

9153 \cs_new_protected:Npn \xeCJK_fntef_sbox:n #1
9154 {
9155   \__xeCJK_boundary_capture_suspend:
9156   \hbox_set:Nn \l__xeCJK_fntef_box
9157   {
9158     \color_ensure_current:
9159     \BeginAccSupp { ActualText = { } }
9160     \cs_if_exist:NT \tag_suspend:n
9161     { \tag_suspend:n { xeCJKfntef } }
9162     #1
9163     \cs_if_exist:NT \tag_resume:n
9164     { \tag_resume:n { xeCJKfntef } }
9165     \EndAccSupp { }
9166   }
9167   \__xeCJK_boundary_capture_resume:
9168 }

```

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。

```

9169 \keys_define:nn { xeCJK / options }
9170 {
9171   underdot / symbol      .tl_set:N = \l__xeCJK_udot_symbol_tl ,
9172   underdot / depth      .tl_set:N = \l__xeCJK_udot_depth_tl ,
9173   underdot / sep        .tl_set:N = \l__xeCJK_udot_sep_tl ,
9174   underdot / format     .tl_set:N = \l__xeCJK_udot_format_tl ,
9175   underdot / textformat .tl_set:N = \l__xeCJK_udot_text_format_tl ,
9176   underdot / boxdepth   .tl_set:N = \l__xeCJK_udot_boxdepth_tl ,
9177   symbol / sep          .tl_set:N = \l__xeCJK_symbol_sep_tl ,
9178   symbol / boxdepth     .tl_set:N = \l__xeCJK_symbol_boxdepth_tl ,
9179   symbol / textformat   .tl_set:N = \l__xeCJK_symbol_text_format_tl ,
9180   underline / skip      .bool_set:N = \l__xeCJK_uline_skip_bool ,

```

```

9181 underline / hidden .bool_set:N = \l__xeCJK_uline_hidden_bool ,
9182 underline / subtract .bool_set:N = \l__xeCJK_uline_subtract_bool ,
9183 underline / thickness .tl_set:N = \l__xeCJK_uline_thickness_tl ,
9184 underline / depth .tl_set:N = \l__xeCJK_uline_depth_tl ,
9185 underline / sep .tl_set:N = \l__xeCJK_uline_sep_tl ,
9186 underline / format .tl_set:N = \l__xeCJK_uline_format_tl ,
9187 underline / textformat .tl_set:N = \l__xeCJK_uline_text_format_tl ,
9188 underdblline / skip .bool_set:N = \l__xeCJK_udbline_skip_bool ,
9189 underdblline / hidden .bool_set:N = \l__xeCJK_udbline_hidden_bool ,
9190 underdblline / subtract .bool_set:N = \l__xeCJK_udbline_subtract_bool ,
9191 underdblline / thickness .tl_set:N = \l__xeCJK_udbline_thickness_tl ,
9192 underdblline / depth .tl_set:N = \l__xeCJK_udbline_depth_tl ,
9193 underdblline / sep .tl_set:N = \l__xeCJK_udbline_sep_tl ,
9194 underdblline / format .tl_set:N = \l__xeCJK_udbline_format_tl ,
9195 underdblline / textformat .tl_set:N = \l__xeCJK_udbline_text_format_tl ,
9196 underdblline / gap .tl_set:N = \l__xeCJK_udbline_gap_tl ,
9197 underwave / skip .bool_set:N = \l__xeCJK_uwave_skip_bool ,
9198 underwave / hidden .bool_set:N = \l__xeCJK_uwave_hidden_bool ,
9199 underwave / subtract .bool_set:N = \l__xeCJK_uwave_subtract_bool ,
9200 underwave / symbol .tl_set:N = \l__xeCJK_uwave_symbol_tl ,
9201 underwave / depth .tl_set:N = \l__xeCJK_uwave_depth_tl ,
9202 underwave / sep .tl_set:N = \l__xeCJK_uwave_sep_tl ,
9203 underwave / format .tl_set:N = \l__xeCJK_uwave_format_tl ,
9204 underwave / textformat .tl_set:N = \l__xeCJK_uwave_text_format_tl ,
9205 sout / skip .bool_set:N = \l__xeCJK_sout_skip_bool ,
9206 sout / hidden .bool_set:N = \l__xeCJK_sout_hidden_bool ,
9207 sout / subtract .bool_set:N = \l__xeCJK_sout_subtract_bool ,
9208 sout / thickness .tl_set:N = \l__xeCJK_sout_thickness_tl ,
9209 sout / height .tl_set:N = \l__xeCJK_sout_height_tl ,
9210 sout / format .tl_set:N = \l__xeCJK_sout_format_tl ,
9211 sout / textformat .tl_set:N = \l__xeCJK_sout_text_format_tl ,
9212 xout / skip .bool_set:N = \l__xeCJK_xout_skip_bool ,
9213 xout / hidden .bool_set:N = \l__xeCJK_xout_hidden_bool ,
9214 xout / subtract .bool_set:N = \l__xeCJK_xout_subtract_bool ,
9215 xout / format .tl_set:N = \l__xeCJK_xout_format_tl ,
9216 xout / textformat .tl_set:N = \l__xeCJK_xout_text_format_tl ,
9217 ulem / skip .bool_set:N = \l__xeCJK_ulem_skip_bool ,
9218 ulem / hidden .bool_set:N = \l__xeCJK_ulem_hidden_bool ,
9219 ulem / subtract .bool_set:N = \l__xeCJK_ulem_subtract_bool ,
9220 ulem / sep .tl_set:N = \l__xeCJK_ulem_sep_tl ,
9221 ulem / boxdepth .tl_set:N = \l__xeCJK_ulem_boxdepth_tl ,
9222 ulem / textformat .tl_set:N = \l__xeCJK_ulem_text_format_tl
9223 }
9224 \clist_map_inline:nn
9225 { underdot , underline , underdblline , underwave , sout , xout , ulem }
9226 {
9227   \keys_define:nn { xeCJK / options }
9228     { #1 .meta:nn = { xeCJK / options / #1 } { ##1 } }
9229 }
9230 \keys_set:nn { xeCJK / options }
9231 {
9232   underdot / symbol = \normalfont . ,
9233   underdot / depth = 0.20 em ,
9234   underdot / sep = 0.04 em ,
9235   symbol / sep = \c_zero_dim ,
9236   underline / skip = true ,
9237   underline / thickness = \ULthickness ,
9238   underline / depth = 0.20 em ,
9239   underline / sep = 0.07 em ,
9240   underdblline / skip = true ,
9241   underdblline / thickness = \ULthickness ,
9242   underdblline / depth = 0.20 em ,
9243   underdblline / sep = 0.17 em ,
9244   underdblline / gap = 1.1 pt ,
9245   underwave / skip = true ,
9246   underwave / symbol = \__xeCJK_fntef_wave_symbol: ,
9247   underwave / depth = 0.20 em ,

```

```

9248     underwave / sep           = 0.00 em ,
9249     sout / skip                = true ,
9250     sout / thickness            = \ULthickness ,
9251     sout / height              = 0.35 em ,
9252     xout / skip                = true
9253 }

```

```

\CJKKunderansymbol 9254 \NewDocumentCommand \CJKKunderansymbol { o m m m }
9255 {
9256     \xeCJK_under_symbol:nnnnnn { symbol } { symbol } {#1} {#2} {#3} {#4}
9257 }

```

\CJKKunderdot 是 \CJKKunderansymbol 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.。

```

9258 \NewDocumentCommand \CJKKunderdot { o m }
9259 {
9260     \xeCJK_under_symbol:nnnnnn { underdot } { udot }
9261     {#1}
9262     { \l__xeCJK_udot_depth_tl }
9263     { \l__xeCJK_udot_format_tl \l__xeCJK_udot_symbol_tl }
9264     {#2}
9265 }

```

\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。

```

9266 \cs_new_protected:Npn \xeCJK_under_symbol:nnnnnn
9267 {
9268     \xeCJK_if_ulem_patch:TF
9269     { \__xeCJK_under_symbol_auxi:nnnnnn }
9270     { \__xeCJK_under_symbol_auxii:nnnnnn }
9271 }
9272 \cs_new_protected:Npn \__xeCJK_under_symbol_auxi:nnnnnn #1#2#3#4#5#6
9273 {
9274     \xeCJK_ulem_right: \UL@stop
9275     \group_begin:
9276     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
9277     \use:c { l__xeCJK_#2_text_format_tl }
9278     \UL@start \xeCJK_ulem_right_node:
9279     #6
9280     \xeCJK_ulem_right: \UL@stop
9281     \group_end:
9282     \UL@start \xeCJK_ulem_right_node:
9283 }
9284 \cs_new_protected:Npn \__xeCJK_under_symbol_auxii:nnnnnn #1#2#3#4#5#6
9285 {
9286     \__xeCJK_boundary_inline_stream_begin:nn { auto } { stream-ulem }
9287     \__xeCJK_boundary_math_begin:n {#6}
9288     \group_begin:
9289     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
9290     \__xeCJK_under_symbol_text_format:c { l__xeCJK_#2_text_format_tl }
9291     #6
9292     \__xeCJK_boundary_math_end:n {#6}
9293     \xeCJK_ulem_right:
9294     \group_end:
9295     \xeCJK_ulem_right_node:
9296     \__xeCJK_boundary_inline_stream_end:n { auto }
9297 }
9298 \cs_new_protected:Npn \xeCJK_under_symbol_initial:nnnnn #1#2#3#4#5
9299 {
9300     \tl_if_novalue:nF {#3}
9301     { \keys_set:nn { xeCJK / options / #1 } {#3} }
9302     \xeCJK_fntef_sbox:n {#5}
9303     \bool_if:NTF \l__xeCJK_fntef_bool
9304     { \xeCJK_make_under_symbol:n { \l__xeCJK_fntef_dim } }
9305     {
9306         \bool_set_true:N \l__xeCJK_fntef_bool

```

```

9307     \xeCJK_make_under_symbol:n {#4}
9308   }
9309   \tl_if_empty:cF { l__xeCJK_#2_boxdepth_tl }
9310   {
9311     \box_set_dp:Nn \l__xeCJK_under_symbol_box
9312     { \use:c { l__xeCJK_#2_boxdepth_tl } }
9313   }
9314   \dim_set:Nn \l__xeCJK_fntef_dim
9315   { \use:c { l__xeCJK_#2_sep_tl } + \box_dp:N \l__xeCJK_under_symbol_box }
9316   \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
9317   \__xeCJK_restore_shipout_CJKsymbol:
9318 }
9319 \cs_new_protected:Npn \__xeCJK_under_symbol_text_format:N #1
9320 {
9321   \tl_if_empty:NF #1
9322   { \xeCJK_ulem_right: #1 \xeCJK_ulem_right_node: }
9323 }
9324 \cs_generate_variant:Nn \__xeCJK_under_symbol_text_format:N { c }
9325 \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。

```

9326 \cs_new_protected:Npn \xeCJK_make_under_symbol:n #1
9327 {
9328   \hbox_set:Nn \l__xeCJK_under_symbol_box
9329   {
9330     \box_move_down:nn { #1 + \box_ht:N \l__xeCJK_fntef_box }
9331     {
9332       \hbox_to_zero:n
9333       {
9334         \xeCJK_select_font:
9335         \tex_kern:D \tex_fontcharwd:D \tex_font:D "4E00 \exp_stop_f:
9336         \tex_hss:D \box_use:N \l__xeCJK_fntef_box \tex_hss:D
9337       }
9338     }
9339   }
9340 }

```

__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。

```

9341 \cs_new_protected:Npn \__xeCJK_restore_shipout_CJKsymbol:
9342 {
9343   \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
9344   { \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N }
9345   \__xeCJK_restore_shipout_fntef:
9346   \xeCJK_cs_clear:N \__xeCJK_restore_shipout_CJKsymbol:
9347 }
9348 \cs_new_protected:Npn \__xeCJK_restore_shipout_fntef:
9349 {
9350   \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
9351   {
9352     \bool_set_false:N \l__xeCJK_fntef_bool
9353     \dim_zero:N \l__xeCJK_fntef_dim
9354   }
9355   \xeCJK_cs_clear:N \__xeCJK_restore_shipout_fntef:
9356 }
9357 \tl_new:N \l__xeCJK_fntef_shipout_tl
9358 \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。

```

9359 \cs_new_protected:Npn \__xeCJK_under_CJKsymbol:N
9360 {
9361   \box_use:N \l__xeCJK_under_symbol_box
9362   \xeCJK_no_break: \__xeCJK_under_CJKsymbol:N
9363 }

```

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

```

9364 \NewDocumentEnvironment { CJKfilltwosides } { 0 { c } m }
9365 {
9366   \use:e { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
9367   \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
9368 }
9369 {
9370   \endminipage
9371   \ignorespacesafterend
9372 }
9373 \NewDocumentEnvironment { CJKfilltwosides* } { 0 { c } m +b }
9374 {
9375   \mode_leave_vertical:
9376   \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
9377   \tl_set:Nn \arraystretch { 1 }
9378   \cs_if_free:NF \extrarowheight
9379   { \cs_set_eq:NN \extrarowheight \c_zero_dim }
9380   \use:e { \_xeCJK_fill_two_sides:nnn {#1} { \dim_eval:n {#2} } } {#3}
9381 }
9382 { \ignorespacesafterend }
9383 \cs_new_protected:Npn \_xeCJK_fill_two_sides:nnn #1#2#3
9384 {
9385   \dim_compare:nNnTF {#2} > \c_zero_dim
9386   {
9387     \hbox_set:Nn \l__xeCJK_tmp_box
9388     { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
9389     \dim_compare:nNnTF {#2} > { \box_wd:N \l__xeCJK_tmp_box }
9390     { \tabular [#1] { @ { } p {#2} @ { } } #3 \endtabular }
9391     { \box_use:N \l__xeCJK_tmp_box }
9392   }
9393   { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
9394 }

```

`\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 `CJKfilltwosides*`。因此, 我们也要用高阶的 `filll`。

```

9395 \cs_new_protected:Npn \xeCJK_fntef_hfilll:
9396 { \skip_horizontal:N \c__xeCJK_filll_skip }
9397 \skip_const:Nn \c__xeCJK_filll_skip { \c_zero_dim plus 1 filll }
9398 </fntef>

```

5.20 xeCJK-listings

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

```

9399 <*listings>
9400 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
9401 \ProcessOptions \scan_stop:
9402 \RequirePackage { xeCJK }
9403 \RequirePackage { listings }
9404 \lst@AddToHook { Init } { \_xeCJK_listings_initial_hook: }
9405 \lst@AddToHook { SelectCharTable } { \_xeCJK_listings_toks_hook: }
9406 \lst@AddToHook { OutputBox }
9407 {
9408   \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
9409   \l__xeCJK_restore_listings_toks_tl
9410   \_xeCJK_listings_output_CM:
9411 }
9412 \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`。

```

9413 \cs_new_protected:Npn \_xeCJK_listings_initial_hook:
9414 {
9415   \tex_noindent:D
9416   \bool_gset_false:N \g__xeCJK_listings_CM_bool
9417   \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
9418   \xeCJK_add_to_shipout:n { \l__xeCJK_restore_listings_toks_tl }
9419   \lst@ifbreaklines
9420     \cs_set_eq:NN \_xeCJK_listings_CJK_toks_hook: \_xeCJK_listings_breaklines_toks:
9421     \tl_if_empty:NF \lst@prebreak
9422       { \tl_put_left:Nn \lst@prebreak { \l__xeCJK_restore_listings_toks_tl } }
9423     \tl_if_empty:NF \lst@postbreak
9424       { \tl_put_left:Nn \lst@postbreak { \l__xeCJK_restore_listings_toks_tl } }
9425   \fi:
9426   \int_set:Nn \l__xeCJK_listings_max_char_int
9427     { \lst@ifec 255 \else: 127 \fi: }
9428 }
9429 \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。

```

9430 \cs_new_protected:Npn \_xeCJK_listings_toks_hook:
9431 {
9432   \tl_clear:N \l__xeCJK_restore_listings_toks_tl
9433   \seq_map_function:NN
9434     \g__xeCJK_class_seq \_xeCJK_backup_inter_class_toks:n
9435     \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
9436     {
9437       \str_if_eq:nnF { ##1 } { Boundary }
9438       {
9439         \xeCJK_inter_class_toks:nnn { Boundary } { ##1 }
9440         { \_xeCJK_listings_process_Default:nN { ##1 } }
9441       }
9442     }
9443   \xeCJK_inter_class_toks:nnn { Boundary } { CM }
9444   { \_xeCJK_listings_process_CM:nN { 0 } }
9445   \_xeCJK_listings_CJK_toks_hook:
9446 }

```

`\_xeCJK_backup_inter_class_toks:n` 注意, 给 `\XeTeXinterchartoks` 赋空值, 会导致 $\text{\XeTeX}$ 崩溃!

```

9447 \cs_new_protected:Npn \_xeCJK_backup_inter_class_toks:n #1
9448 {
9449   \tl_set:Ne \l__xeCJK_tmp_tl
9450   { \xeCJK_get_inter_class_toks:nn { Boundary } {#1} }
9451   \tl_put_right:Ne \l__xeCJK_restore_listings_toks_tl
9452   {
9453     \xeCJK_inter_class_toks:nnn { Boundary } {#1}
9454     {
9455       \tl_if_empty:NTF \l__xeCJK_tmp_tl
9456         { \exp_not:N \prg_do_nothing: }
9457         { \exp_not:o \l__xeCJK_tmp_tl }
9458     }
9459   }
9460 }
9461 \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:`

```

9462 \cs_new_protected:Npn \_xeCJK_listings_CJK_toks_hook:
9463 {

```

```

9464 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
9465 { \__xeCJK_listings_process_CJK:nN { 2 } }
9466 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
9467 { \__xeCJK_listings_process_CJK:nN { 2 } }
9468 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
9469 { \__xeCJK_listings_process_CJK:nN { 2 } }
9470 \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
9471 { \__xeCJK_listings_process_CJK:nN { 2 } }
9472 \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
9473 { \__xeCJK_listings_process_CJK:nN { 2 } }
9474 \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
9475 { \__xeCJK_listings_process_CM:nN { 0 } }
9476 \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
9477 { \__xeCJK_listings_process_CM:nN { 0 } }
9478 \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
9479 { \__xeCJK_listings_process_CJK:nN { 2 } }
9480 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
9481 {
9482   \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
9483   { \__xeCJK_listings_process_CJK:nN { 2 } }
9484 }
9485 }
9486 \cs_new_protected:Npn \__xeCJK_listings_breaklines_toks:
9487 {
9488   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
9489   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9490   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
9491   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9492   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
9493   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9494   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
9495   { \__xeCJK_listings_process_CM:nN { 0 } }
9496   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
9497   { \__xeCJK_listings_process_CM:nN { 0 } }
9498   \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
9499   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9500   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
9501   { \__xeCJK_listings_process_FullLeft:nN { 2 } }
9502   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
9503   { \__xeCJK_listings_process_FullRight:nN { 2 } }
9504   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
9505   {
9506     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
9507     { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
9508   }
9509 }

```

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

```

9510 \cs_new_protected:Npn \__xeCJK_listings_process_Default:nN #1#2
9511 {
9512   \int_compare:nNnTF
9513   { \xeCJK_token_value_charcode:N #2 } > \l__xeCJK_listings_max_char_int
9514   {
9515     \token_if_letter:NTF #2
9516     { \lst@ProcessLetter #2 }
9517     { \lst@ProcessOther #2 }
9518   }
9519   { \__xeCJK_listings_output_Default:nN {#1} #2 }
9520 }

```

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:。

```

9521 \cs_new_protected:Npn \__xeCJK_listings_output_Default:nN #1#2
9522 {
9523   \group_begin:
9524   \xeCJK_clear_inter_class_toks:nn { Boundary } {#1}

```

```

9525     \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \group_end: }
9526     #2
9527     \scan_stop:
9528 }

```

_xeCJK_listings_process_CJK:nN 对 CJK 字符类的处理。

```

9529 \cs_new_protected:Npn \_xeCJK_listings_process_CJK:nN #1#2
9530 {
9531     \token_if_letter:NTF #2
9532     { \_xeCJK_listings_process_letter:nN {#1} #2 }
9533     { \_xeCJK_listings_process_other:nN {#1} #2 }
9534 }

```

_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。

```

9535 \cs_new_protected:Npn \_xeCJK_listings_append:nN #1#2
9536 {
9537     \int_add:Nn \lst@length { #1 - 1 }
9538     \lst@Append #2
9539 }

```

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

_xeCJK_listings_process_other:nN

```

9540 \cs_new_protected:Npn \_xeCJK_listings_process_letter:nN
9541 {
9542     \lst@whitespacefalse
9543     \bool_if:NTF \l__xeCJK_listings_letter_bool
9544     { \lst@lettertrue }
9545     {
9546         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9547         \bool_set_true:N \l__xeCJK_listings_letter_bool
9548     }
9549     \_xeCJK_listings_append:nN
9550 }
9551 \bool_new:N \l__xeCJK_listings_letter_bool
9552 \cs_new_protected:Npn \_xeCJK_listings_process_other:nN #1#2
9553 {
9554     \lst@whitespacefalse
9555     \bool_if:NTF \l__xeCJK_listings_letter_bool
9556     {
9557         \lst@Output \lst@letterfalse
9558         \bool_set_false:N \l__xeCJK_listings_letter_bool
9559     }
9560     { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
9561     \cs_set_eq:NN \lst@lastother #2
9562     \_xeCJK_listings_append:nN {#1} #2
9563 }

```

_xeCJK_listings_process_breaklines_CJK:nN 当使用 breaklines 选项时,立即输出之前的单个文字,以便于断行。并将标点与它前/后的 CJK 文字放在同一个盒子中,以保持禁则。但是不能区分 letter 和 other。

_xeCJK_listings_process_FullLeft:nN

_xeCJK_listings_process_FullRight:nN

```

9564 \cs_new_protected:Npn \_xeCJK_listings_process_breaklines_CJK:nN
9565 {
9566     \lst@whitespacefalse
9567     \bool_if:NTF \l__xeCJK_listings_letter_bool
9568     {
9569         \int_compare:nNnF \l__xeCJK_listings_flag_int = 2 { \lst@Output }
9570         \lst@lettertrue
9571     }
9572     {
9573         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9574         \bool_set_true:N \l__xeCJK_listings_letter_bool
9575     }
9576     \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one_int

```

```

9577     \__xeCJK_listings_append:nN
9578   }
9579   \cs_new_protected:Npn \__xeCJK_listings_process_FullLeft:nN #1#2
9580   {
9581     \lst@whitespacefalse
9582     \bool_if:NTF \l__xeCJK_listings_letter_bool
9583     {
9584       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2
9585       {
9586         \int_compare:nNnTF \l__xeCJK_listings_flag_int = 3
9587         { \bool_if:NT \l__xeCJK_punct_breakable_bool { \lst@Output } }
9588         { \lst@Output }
9589       }
9590       \lst@lettertrue
9591     }
9592     {
9593       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9594       \bool_set_true:N \l__xeCJK_listings_letter_bool
9595     }
9596     \int_set:Nn \l__xeCJK_listings_flag_int { 2 }
9597     \__xeCJK_listings_append:nN {#1} #2
9598   }
9599   \cs_new_protected:Npn \__xeCJK_listings_process_FullRight:nN #1#2
9600   {
9601     \lst@whitespacefalse
9602     \bool_if:NTF \l__xeCJK_listings_letter_bool
9603     {
9604       \int_compare:nNnT \l__xeCJK_listings_flag_int < 2
9605       { \__xeCJK_punct_if_long:NT #2 { \lst@Output } }
9606       \lst@lettertrue
9607     }
9608     {
9609       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
9610       \bool_set_true:N \l__xeCJK_listings_letter_bool
9611     }
9612     \int_set:Nn \l__xeCJK_listings_flag_int { 3 }
9613     \__xeCJK_listings_append:nN {#1} #2
9614   }
9615   \int_new:N \l__xeCJK_listings_flag_int
\lst@AppendLetter 9616 \cs_set_protected:Npn \lst@AppendLetter
\lst@AppendOther 9617 {
9618   \bool_if:NTF \l__xeCJK_listings_letter_bool
9619   {
9620     \lst@Output \lst@lettertrue
9621     \bool_set_false:N \l__xeCJK_listings_letter_bool
9622   }
9623   { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
9624   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
9625   \lst@Append
9626 }
9627 \cs_set_protected:Npn \lst@AppendOther
9628 {
9629   \bool_if:NTF \l__xeCJK_listings_letter_bool
9630   {
9631     \lst@Output \lst@letterfalse
9632     \bool_set_false:N \l__xeCJK_listings_letter_bool
9633   }
9634   { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
9635   \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
9636   \tex_futurelet:D \lst@lastother \lst@Append
9637 }
\__xeCJK_listings_process_CM:nN CM 类作为 letter 处理, 不用增加 \lst@length。
9638 \cs_new_protected:Npn \__xeCJK_listings_process_CM:nN
9639 {
9640   \reverse_if:N \lst@ifflexible

```

```

9641     \bool_gset_true:N \g__xeCJK_listings_CM_bool
9642     \fi:
9643     \__xeCJK_listings_process_letter:nN
9644   }

```

__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 $\text{\TeX}$ 将基本字和组合标识正确的组合起来。

```

9645 \cs_new_protected:Npn \__xeCJK_listings_output_CM:
9646 {
9647   \reverse_if:N \lst@ifflexible
9648   \bool_if:NT \g__xeCJK_listings_CM_bool
9649   {
9650     \bool_gset_false:N \g__xeCJK_listings_CM_bool
9651     \xeCJK_cs_clear:N \lst@FillOutputBox
9652     \cs_set_eq:NN \CJKglue \tex_hss:D
9653   }
9654   \fi:
9655 }
9656 \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。

```

9657 \cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
9658 {
9659   \token_if_active:NTF #3
9660   { #1#3 }
9661   {
9662     \token_if_cs:NTF #3
9663     { #2#3 }
9664     {
9665       \int_compare:nNnTF { `#3 } > { \l__xeCJK_listings_max_char_int }
9666       { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
9667       { #2#3 }
9668     }
9669   }
9670 }
9671 \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, $\text{\TeX}$ 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)。

```

9672 \cs_new_protected:Npn \__xeCJK_listings_rescan:Nn #1#2
9673 {
9674   \__xeCJK_listings_set_escape:
9675   \tl_set:Nn \l__xeCJK_tmp_tl {#2}
9676   \__xeCJK_listings_escape_backslash:
9677   \tl_clear:N \l__xeCJK_tmpb_tl
9678   \tl_map_inline:Nn \l__xeCJK_tmp_tl
9679   {
9680     \token_if_parameter:NTF ##1
9681     {
9682       \tl_put_right:Ne \l__xeCJK_tmpb_tl
9683       {

```

```

9684         \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:N
9685         \char_generate:nn { \int_value:w `##1 } { 13 }
9686     }
9687 }
9688 { \tl_put_right:Nn \l__xeCJK_tmpb_tl { ##1 } }
9689 }
9690 \tl_set_eq:NN \l__xeCJK_tmp_tl \l__xeCJK_tmpb_tl
9691 \tl_set_rescan:Nno #1
9692 { \char_set_catcode_active:n { 35 } }
9693 { \l__xeCJK_tmp_tl }
9694 }
9695 \cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
9696 {
9697     \__xeCJK_listings_rescan:Nn \l__xeCJK_tmp_tl {#1}
9698     \tl_put_right:No \lst@arg { \l__xeCJK_tmp_tl }
9699 }
9700 \cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
9701 \cs_new_protected:Npn \__xeCJK_listings_inline_group:w
9702 {
9703     \exp_after:wN \__xeCJK_listings_inline_group:n
9704     \exp_after:wN { \if_int_compare:w `} = \c_zero_int \fi:
9705 }
9706 \cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
9707 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
9708 {
9709     \__xeCJK_listings_rescan:Nn \lst@arg {#1}
9710     \lst@InlineGJEnd
9711 }

```

__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。

```

9712 \clist_map_inline:nn { \lstinline , \lstinline@ , \lst@InlineG , \lst@DeInit }
9713 { \__xeCJK_boundary_reserve:n {#1} }
9714 \bool_new:N \l__xeCJK_listings_boundary_bool
9715 \cs_new_protected:Npn \__xeCJK_listings_boundary_stream_end:
9716 { \__xeCJK_boundary_inline_stream_end:n { auto } }
9717 \cs_new_eq:NN \__xeCJK_orig_listings_inline:n \lstinline@
9718 \cs_set_protected:Npn \lstinline@ #1
9719 {
9720     \__xeCJK_boundary_inline_stream_begin:n { auto }
9721     \bool_set_true:N \l__xeCJK_listings_boundary_bool
9722     \__xeCJK_orig_listings_inline:n #1
9723 }
9724 \cs_new_eq:NN \__xeCJK_orig_listings_inline_group: \lst@InlineG
9725 \cs_set_protected:Npn \lst@InlineG
9726 {
9727     \__xeCJK_boundary_inline_stream_begin:n { auto }
9728     \bool_set_true:N \l__xeCJK_listings_boundary_bool
9729     \__xeCJK_orig_listings_inline_group:
9730 }
9731 \tl_gput_right:Nn \lst@DeInit
9732 {
9733     \bool_if:NT \l__xeCJK_listings_boundary_bool
9734     { \aftergroup \__xeCJK_listings_boundary_stream_end: }
9735 }

```

__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。

```

9736 \group_begin:
9737 \cs_set:Npn \__xeCJK_tmp:w #1
9738 {

```

```

9739 \group_end:
9740 \cs_new_protected:Npn \__xeCJK_listings_set_escape:
9741 { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
9742 \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
9743 { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
9744 }
9745 \use:n
9746 {
9747 \char_set_catcode_active:N \
9748 \__xeCJK_tmp:w
9749 }
9750 { \ }

```

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

```

9751 \cs_new_protected:Npe \__xeCJK_listings_escape_backslash:
9752 {
9753 \tl_replace_all:Nnn \exp_not:N \l__xeCJK_tmp_tl
9754 { \c_backslash_str }
9755 { \c_backslash_str \c_backslash_str }
9756 }
9757 </listings>
9758 <@@=xunadd>

```

5.21 xunicode-addon

```
9759 <*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。

```

9760 \bool_lazy_or:nnF
9761 { \sys_if_engine_xetex_p: }
9762 { \sys_if_engine luatex_p: }
9763 {
9764 \msg_new:nnnn { xunicode-addon } { xetex-luatex }
9765 { This~package~requires~either~XeTeX~or~LuaTeX~to~function.}
9766 {
9767 You~must~change~your~typesetting~engine~to,~e.g.,\
9768 "xelatex"~or~"lualatex"~instead~of~plain~"latex"~or~"pdflatex".
9769 }
9770 \msg_critical:nn { xunicode-addon } { xetex-luatex }
9771 }
9772 \RequirePackage { xparse }

```

宏包选项是编码的名字。

```

9773 \clist_new:N \g__xunadd_encname_clist
9774 \tl_if_exist:NT \UTFencname
9775 { \clist_gput_right:Ne \g__xunadd_encname_clist { \UTFencname } }
9776 \DeclareOption*
9777 { \clist_gput_right:No \g__xunadd_encname_clist \CurrentOption }
9778 \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 `LuaLaTeX`，则需要作一些设置，使得 `xunicode` 可用。

```

9779 \@ifpackageloaded { xunicode } { }
9780 {
9781 \clist_get:NNF \g__xunadd_encname_clist \UTFencname

```

```

9782     {
9783       \cs_if_exist:NTF \UnicodeEncodingName
9784       { \tl_set:Nn \UTFencname { \UnicodeEncodingName } }
9785       {
9786         \sys_if_engine_xetex:TF
9787         { \tl_set:Nn \UTFencname { EU1 } }
9788         { \tl_set:Nn \UTFencname { EU2 } }
9789       }
9790       \clist_gset_eq:NN \g__xunadd_encname_clist \UTFencname
9791     }
9792     \sys_if_engine_xetex:TF
9793     { \RequirePackage { xunicode } }
9794     {
9795       \cs_set_eq:NN \__xunadd_tmp:w \XeTeXpicfile
9796       \cs_set_eq:NN \XeTeXpicfile \prg_do_nothing:
9797       \RequirePackage { xunicode }
9798       \cs_set_eq:NN \XeTeXpicfile \__xunadd_tmp:w
9799     }
9800   }
9801   \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。

```

9802 \RenewDocumentCommand \ReloadXunicode { m }
9803 {
9804   \clist_set:Nn \l__xunadd_encname_clist {#1}
9805   \__xunadd_reload:N \l__xunadd_encname_clist
9806 }
9807 \cs_new_protected:Npn \__xunadd_reload:N #1
9808 {
9809   \cs_set_eq:NN \__xunadd_tmp:w \iftipaonetoken
9810   \cs_set_eq:NN \iftipaonetoken \scan_stop:
9811   \use:e
9812   {
9813     \ExplSyntaxOff
9814     \char_set_catcode_letter:n { 64 }
9815     \exp_not:N \clist_map_function:NN \exp_not:N #1 \__xunadd_reload_aux:n
9816     \bool_if:NTF \l_kernel_expl_bool
9817     { \ExplSyntaxOn }
9818     { \ExplSyntaxOff }
9819     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
9820   }
9821   \cs_set_eq:NN \iftipaonetoken \__xunadd_tmp:w
9822 }
9823 \cs_new_protected:Npn \__xunadd_reload_aux:n #1
9824 {
9825   \cs_if_exist:cTF { T@ #1 }
9826   {
9827     \tl_set:Nn \UTFencname {#1}
9828     \clist_if_in:NnF \g__xunadd_encname_clist {#1}
9829     { \clist_gput_right:Nn \g__xunadd_encname_clist {#1} }
9830     \file_input:n { xunicode.sty }
9831     \file_input:n { xunicode-extra.def }
9832   }
9833   { \msg_error:nnn { xunicode-addon } { encoding-unknown } {#1} }
9834 }
9835 \clist_new:N \l__xunadd_encname_clist
9836 \msg_new:nnnn { xunicode-addon } { encoding-unknown }
9837 { Encoding~scheme~"#1"~unknown. }
9838 {
9839   You~may~use \\\
9840   \token_to_str:N \usepackage [ #1 , \encodingdefault ] \{fontenc\} \\\
9841   before~xunicode-addon~or~xunicode.
9842 }

```

`\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。

```

9843 \RenewDocumentCommand \DeclareUTFmathsymbols { m }
9844 {
9845   \bool_if:NT \l__xunadd_math_as_UTF_text_bool
9846   {
9847     \seq_map_inline:Nn \l__xunadd_math_as_UTF_text_seq
9848     { \__xunadd_declare_math_as_UTF_text:n {##1} }
9849     \bool_set_false:N \l__xunadd_math_as_UTF_text_bool
9850   }
9851 }
9852 \seq_new:N \l__xunadd_math_as_UTF_text_seq
9853 \seq_set_from_clist:Nn \l__xunadd_math_as_UTF_text_seq
9854 { hbar , Finv , aleph , beth , gimel , daleth , Game }
9855 \bool_new:N \l__xunadd_math_as_UTF_text_bool
9856 \RenewDocumentCommand \UseMathAsText { }
9857 {
9858   \math@s@text@true
9859   \bool_set_true:N \l__xunadd_math_as_UTF_text_bool
9860 }
9861 \@onlypreamble \UseMathAsText
9862 \cs_new_protected:Npn \__xunadd_declare_math_as_UTF_text:n #1
9863 {
9864   \cs_if_exist:cTF {#1}
9865   {
9866     \cs_new_eq:cc { keepmathUTF #1 } {#1}
9867     \cs_gset_protected:cpe {#1}
9868     {
9869       \exp_not:N \mode_if_math:TF
9870       { \exp_not:c { keepmathUTF #1 } }
9871       { \exp_not:c { text #1 } }
9872     }
9873     \tl_put_right:Ne \l__xunadd_hyperref_hook_tl
9874     { \cs_set_eq:NN \exp_not:c {#1} \exp_not:c { text #1 } }
9875   }
9876   { \cs_new:cpe {#1} { \exp_not:c { text #1 } } }
9877 }
9878 \tl_new:N \l__xunadd_hyperref_hook_tl
9879 \AtBeginDocument
9880 {
9881   \cs_if_free:NF \pdfstringdefDisableCommands
9882   { \pdfstringdefDisableCommands { \l__xunadd_hyperref_hook_tl } }
9883 }

```

`\__xunadd_glyph_if_exist:p:n` 判断字符在当前字体中是否存在。

```

\__xunadd_glyph_if_exist:nTF
9884 \prg_new_conditional:Npnn \__xunadd_glyph_if_exist:n #1 { p , T , F , TF }
9885 {
9886   \tex_iffontchar:D \tex_font:D \tex_numexpr:D #1 \scan_stop:
9887   \prg_return_true: \else: \prg_return_false: \fi:
9888 }

```

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

```

9889 \RenewDocumentCommand \UndeclareUTFcharacter { O { \UTFencname } m m }
9890 {
9891   \__xunadd_if_csname:nTF {#3}
9892   { \UndeclareTextCommand {#3} }
9893   { \exp_args:Nc \UndeclareTextCommand { \tl_to_str:n {#3} } }
9894   {#1}
9895 }

```

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

```

9896 \RenewDocumentCommand \UndeclareUTFcomposite { O { \UTFencname } m m m }
9897 {
9898   \__xunadd_if_csname:nTF {#3}
9899   { \__xunadd_undeclare_composite:Nnnn #3 }
9900   { \exp_args:Nc \__xunadd_undeclare_composite:Nnnn { \tl_to_str:n {#3} } }

```

```

9901      {#1} {#4} {#2}
9902    }
9903 \cs_new_protected:Npn \__xunadd_undeclare_composite:Nnnn #1#2#3#4
9904 { \cs_undefine:c { \__xunadd_composite_cs:Nnn #1 {#2} {#3} } }

\__xunadd_composite_cs:Nnn 9905 \cs_new:Npe \__xunadd_composite_cs:Nnn #1#2#3
\__xunadd_composite_cs:nnn 9906 { \c_backslash_str #2 \exp_not:N \token_to_str:N #1 - \exp_not:N \tl_to_str:n {#3} }
9907 \cs_new:Npe \__xunadd_composite_cs:nnn #1#2#3
9908 { \c_backslash_str #2 #1 - \exp_not:N \tl_to_str:n {#3} }

```

__xunadd_if_csname:nTF 判断 #1 是否可以作为控制序列的名字。这是因为 xunicode 使用了下面的定义。

```

\DeclareUTFcharacter[\UTFencname]{x0149}{'n}

9909 \prg_new_conditional:Npnn \__xunadd_if_csname:n #1 { TF }
9910 {
9911   \tl_if_single_token:nTF {#1}
9912   {
9913     \token_if_cs:NTF #1
9914     { \prg_return_true: }
9915     {
9916       \token_if_active:NTF #1
9917       { \prg_return_true: }
9918       { \prg_return_false: }
9919     }
9920   }
9921   { \prg_return_false: }
9922 }

```

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

```

9923 \RenewDocumentCommand \DeclareUTFcharacter { 0 { \UTFencname } m m }
9924 {
9925   \cs_if_exist_use:cF
9926   { __xunadd_restore_ \tl_to_str:n {#3} : }
9927   {
9928     \__xunadd_if_csname:nTF {#3}
9929     { \__xunadd_declare_character:Nnn #3 }
9930     { \__xunadd_declare_character:cnn { \tl_to_str:n {#3} } }
9931     {#1} {#2}
9932   }
9933 }

```

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

```

9934 \cs_new_protected:cpn
9935 { __xunadd_restore_ \tl_to_str:n { \hbar } : }
9936 { \__xunadd_restore_cmd:N \hbar }
9937 \cs_new_protected:cpn
9938 { __xunadd_restore_ \tl_to_str:n { \nobreakspace } : }
9939 { \__xunadd_restore_cmd:N \nobreakspace }
9940 \cs_new_protected:Npn \__xunadd_restore_cmd:N #1
9941 { \__xunadd_restore_cmd:Ne #1 { ? - \token_to_str:N #1 } }
9942 \cs_new_protected:Npn \__xunadd_restore_cmd:Nn #1#2
9943 {
9944   \cs_if_free:cF {#2}
9945   { \__xunadd_restore_cmd:Nc #1 {#2} }
9946 }
9947 \cs_new_protected:Npn \__xunadd_restore_cmd:NN #1#2
9948 {
9949   \cs_gset_eq:NN #1 #2
9950   \cs_undefine:N #2
9951 }
9952 \cs_generate_variant:Nn \__xunadd_restore_cmd:Nn { Ne }
9953 \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。

```

9954 \cs_new_protected:Npn \__xunadd_declare_character:Nnn #1#2#3

```

```

9955 {
9956   \__xunadd_provide_text_command_default:N #1
9957   \exp_after:wN \__xunadd_declare_character:Nn
9958   \tex_Uchar:D \__xunadd_check_slot:n {#3} \exp_stop_f:
9959   #1 { \token_to_str:N #1 } {#2}
9960 }
9961 \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

```

9962 \NewDocumentCommand \DeclareUTFSymbol { m O { \UTFencname } m }
9963 { \__xunadd_declare_character:Nnn #1 {#2} {#3} }
9964 \NewDocumentCommand \DeclareUTFCommand { m O { \UTFencname } m }
9965 { \__xunadd_text_command:Nonn #1 { \token_to_str:N #1 } {#2} {#3} }
9966 \cs_new_protected:Npn \__xunadd_text_command:Nnnn #1#2#3#4
9967 { \DeclareTextCommand #1 {#3} { \__xunadd_text_command:nn {#2} {#4} } }
9968 \cs_generate_variant:Nn \__xunadd_text_command:Nnnn { No }
9969 \cs_new_protected:Npn \__xunadd_text_command:nn #1#2
9970 {
9971   \__xunadd_begin_hook:nn {#1} {#2}
9972   #2
9973   \__xunadd_end_hook:nn {#1} {#2}
9974 }

```

__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:->\texttrhookbelow {\textopeno }
\ hbar macro:->{\mathchar '26\mkern -9mu}
\ textaolig macro:->{a\kern -.25em o}

```

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

```

9975 \cs_new_protected:Npn \__xunadd_provide_text_command_default:N #1
9976 {
9977   \cs_if_exist:cF { ? \token_to_str:N #1 }
9978   {
9979     \cs_if_free:cF { ? - \token_to_str:N #1 }
9980     {
9981       \exp_args:NNv \ProvideTextCommandDefault #1
9982       { ? - \token_to_str:N #1 }
9983     }
9984   }
9985 }

```

__xunadd_declare_character:Nnn 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。

```

9986 \cs_new_protected:Npn \__xunadd_declare_character:Nnnn #1#2#3#4
9987 { \DeclareTextCommand #2 {#4} { \__xunadd_text_character:nN {#3} {#1} } }
9988 \cs_new_protected:Npn \__xunadd_text_character:nN #1#2
9989 {
9990   \__xunadd_begin_hook:nn {#1} {#2}
9991   \__xunadd_glyph_if_exist:nTF { `#2 }
9992   {#2} { \cs_if_exist_use:cF { ? #1 } {#2} }
9993   \__xunadd_end_hook:nn {#1} {#2}
9994 }
9995 \cs_generate_variant:Nn \__xunadd_declare_character:Nnnn { NNe }

```

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

```

9996 \cs_new:Npn \__xunadd_check_slot:n #1

```

```

9997 {
9998   \int_eval:n
9999   {
10000     \tl_if_head_eq_charcode:nNTF {#1} x
10001     { " \use_none:n #1 } {#1}
10002   }
10003 }

```

`\DeclareUTFcomposite` 设置编码 #1 下的符号命令 #3 与它的参数 #4 的复合对应的符号的 Unicode 是 #2。

```

10004 \RenewDocumentCommand \DeclareUTFcomposite { 0 { \UTFencname } m m m }
10005 {
10006   \__xunadd_if_csname:nTF {#3}
10007   { \__xunadd_declare_composite:Nnnn #3 }
10008   { \__xunadd_declare_composite:cnnn { \tl_to_str:n {#3} } }
10009   {#1} {#4} {#2}
10010 }

```

`\__xunadd_declare_composite:Nnnn` 这里使用 `\tex_afterassignment:D` 是因为 xunicode 有如下的定义。

```

\DeclareUTFcomposite[\UTFencname]{x02E8\char"02E5}{\tonebar}{25}
\DeclareUTFcomposite[\UTFencname]{x02E5\char"02E8}{\tonebar}{52}

```

对复合符号命令的定义用的是 `\chardef`, 这有利于下面字符是否存在的判断。

```

10011 \cs_new_protected:Npn \__xunadd_declare_composite:Nnnn #1#2#3#4
10012 {
10013   \tex_afterassignment:D \use_none_delimit_by_q_stop:w
10014   \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
10015   { \__xunadd_check_slot:n {#4} }
10016   \q_stop
10017 }
10018 \cs_new_protected:Npn \__xunadd_chardef:Nn #1#2
10019 { \tex_chardef:D #1 = \tex_numexpr:D #2 \scan_stop: }
10020 \cs_generate_variant:Nn \__xunadd_chardef:Nn { c }
10021 \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。

```

10022 \NewDocumentCommand \DeclareUTFCompositeCommand { m 0 { \UTFencname } m m }
10023 { \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。

```

10024 \NewDocumentCommand \DeclareUTFCompositeSymbol { m 0 { \UTFencname } m m }
10025 {
10026   \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
10027   { \__xunadd_check_slot:n {#4} }
10028 }

```

`\DeclareUTFComposite` 将 #1 设置为编码 #2 下的带一个参数的复合符号命令。

```

10029 \NewDocumentCommand \DeclareUTFComposite { m 0 { \UTFencname } }
10030 { \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。

```

10031 \NewDocumentCommand \DeclareUTFEncodedAccent { m 0 { \UTFencname } m m }
10032 { \__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。

```

10033 \NewDocumentCommand \DeclareUTFEncodedAccents { m 0 { \UTFencname } m m }
10034 { \__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。

```
10035 \NewDocumentCommand \DeclareUTFEncodedSymbol { m O { \UTFencname } m m }
10036 { \__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。

```
10037 \NewDocumentCommand \DeclareUTFEncodedCircle { m O { \UTFencname } m m }
10038 { \__xunadd_declare_encoded:NNnnn \__xunadd_combine_circle:nnNNn #1 {#2} {#3} {#4} }
```

`\DeclareEncodedCompositeCharacter` 10039 \RenewDocumentCommand \DeclareEncodedCompositeCharacter { m m m m }
10040 { \DeclareUTFEncodedSymbol #2 [#1] { "#3 } { "0#4 } }

`\DeclareEncodedCompositeAccents` 10041 \RenewDocumentCommand \DeclareEncodedCompositeAccents { m m m m }
10042 { \DeclareUTFEncodedAccents #2 [#1] { "#4 } { "#3 } }

`\DeclareUTFDoubleEncodedAccent` 10043 \NewDocumentCommand \DeclareUTFDoubleEncodedAccent { m O { \UTFencname } m m }
10044 { __xunadd_declare_encoded:NNnnn __xunadd_combine_double_accent:nnNNn #1 {#2} {#3} {#4} }

`\DeclareUTFDoubleEncodedSymbol` 10045 \NewDocumentCommand \DeclareUTFDoubleEncodedSymbol { m O { \UTFencname } m m }
10046 { __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。

```
10047 \cs_new_protected:Npn \__xunadd_declare_composite:Nnn #1#2#3
10048 { \DeclareTextCommand #1 {#3} { \__xunadd_text_composite:nnn {#2} {#3} } }
```

```
\__xunadd_text_composite:nnn 10049 \cs_new_protected:Npn \__xunadd_text_composite:nnn #1#2#3
10050 {
10051   \__xunadd_begin_hook:nn {#1} {#3}
10052   \cs_if_exist:cTF { \__xunadd_composite_cs:nnn {#1} {#2} {#3} }
10053   {
10054     \__xunadd_text_composite:cnn
10055     { \__xunadd_composite_cs:nnn {#1} {#2} {#3} } {#1} {#3}
10056   }
10057   { \cs_if_exist_use:cTF { ? #1 } { {#3} } {#3} }
10058   \__xunadd_end_hook:nn {#1} {#3}
10059 }
10060 \cs_new_protected:Npn \__xunadd_text_composite:Nnn #1#2#3
10061 {
10062   \token_if_chardef:NTF #1
10063   {
10064     \__xunadd_glyph_if_exist:nTF {#1}
10065     {#1} { \cs_if_exist_use:cTF { ? #2 } { {#3} } {#3} }
10066   }
10067   {#1}
10068 }
10069 \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。

```
10070 \cs_new_protected:Npn \__xunadd_declare_encoded:NNnnn #1#2#3#4#5
10071 {
10072   \exp_after:wN \__xunadd_declare_encoded:NNNNee
10073   \tex_Uchar:D \__xunadd_check_slot:n {#4} \exp_after:wN \exp_stop_f:
10074   \tex_Uchar:D \__xunadd_check_slot:n {#5} \exp_stop_f:
10075   #1 #2 { \token_to_str:N #2 } {#3}
10076 }
10077 \cs_new_protected:Npn \__xunadd_declare_encoded:NNNNnn #1#2#3#4#5#6
10078 { \DeclareTextCommand #4 {#6} { #3 {#5} {#6} {#1} {#2} } }
10079 \cs_generate_variant:Nn \__xunadd_declare_encoded:NNnnn { c }
10080 \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。

```
10081 \cs_new_protected:Npn \__xunadd_text_combine:NnnNNn #1#2#3#4#5#6
```

```

10082 {
10083   \__xunadd_begin_hook:nn {#2} {#6}
10084   \cs_if_exist:cTF { \__xunadd_composite_cs:nnn {#2} {#3} {#6} }
10085   {
10086     \__xunadd_text_combine:cNnNNn
10087     { \__xunadd_composite_cs:nnn {#2} {#3} {#6} } #1 {#2} {#4} {#5} {#6}
10088   }
10089   { #1 {#6} {#2} {#4} {#5} }
10090   \__xunadd_end_hook:nn {#2} {#6}
10091 }
10092 \cs_new_protected:Npn \__xunadd_text_combine:NNnNNn #1#2#3#4#5#6
10093 {
10094   \token_if_chardef:NTF #1
10095   { \__xunadd_glyph_if_exist:nTF {#1} {#1} { #2 {#6} {#3} {#4} {#5} } }
10096   {#1}
10097 }
10098 \cs_generate_variant:Nn \__xunadd_text_combine:NNnNNn { c }

\__xunadd_combine_symbol:nnNNn 10099 \cs_new_protected:Npn \__xunadd_combine_symbol:nnNNn
10100 { \__xunadd_text_combine:NnnNNn \__xunadd_add_symbol:nnNN }
10101 \cs_new_protected:Npn \__xunadd_add_symbol:nnNN #1#2#3#4
10102 {
10103   \tl_if_blank:nTF {#1}
10104   {
10105     \__xunadd_glyph_if_exist:nTF { `#4 }
10106     {#4}
10107     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10108   }
10109   {
10110     \__xunadd_glyph_if_exist:nTF { `#3 }
10111     { #1#3 }
10112     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10113   }
10114 }

```

若组合重音符号的 #3 和基本重音符号 #4 在当前字体中都不存在，则转换到 \DeclareTextAccentDefault 设置的编码或者使用 \DeclareTextCommandDefault 中设置的命令。0.9999 版以前的 XeTeX 需要设置 \XeTeXinputnormalization 为 1，才能使用字体中由基础字符和组合符号对应的实际字符；而 0.9999 版以后的 XeTeX 默认就启用这个功能，\XeTeXinputnormalization 似乎是无效的，怀疑是使用 HarfBuzz 库替代 ICU 进行字体排版的缘故¹⁹。

```

10115 \cs_new_protected:Npn \__xunadd_combine_accent:nnNNn
10116 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accent:nnNN }
10117 \cs_new_protected:Npn \__xunadd_add_accent:nnNN #1#2#3#4
10118 {
10119   \tl_if_blank:nTF {#1}
10120   {
10121     \__xunadd_glyph_if_exist:nTF { `#4 }
10122     {#4}
10123     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10124   }
10125   {
10126     \__xunadd_glyph_if_exist:nTF { `#3 }
10127     { #1#3 }
10128     {
10129       \__xunadd_glyph_if_exist:nTF { `#4 }
10130       { \add@accent { `#4 } {#1} }
10131       { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10132     }
10133   }
10134 }

```

```

\__xunadd_combine_accents:nnNNn 10135 \cs_new_protected:Npn \__xunadd_combine_accents:nnNNn
\__xunadd_add_accents:nnNN

```

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

```

10136 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accents:nnNN }
10137 \cs_new_protected:Npn \__xunadd_add_accents:nnNN #1#2#3#4
10138 {
10139   \tl_if_blank:nTF {#1}
10140     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
10141     {
10142       \__xunadd_glyph_if_exist:nTF { `#3 }
10143       { \__xunadd_glyph_if_exist:nTF { `#4 } }
10144       { \use_ii:nn }
10145       { #1#3#4 }
10146       { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3#4 } }
10147     }
10148 }

```

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 L^AT_EX 中的设置。

```

10149 \cs_new_protected:Npn \__xunadd_combine_circle:nnNNn
10150 { \__xunadd_text_combine:NnnNNn \__xunadd_add_circle:nnNN }
10151 \cs_new_protected:Npn \__xunadd_add_circle:nnNN #1#2#3#4
10152 {
10153   \tl_if_blank:nTF {#1}
10154   {
10155     \__xunadd_glyph_if_exist:nTF { `#4 }
10156     {#4}
10157     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10158   }
10159   {
10160     \__xunadd_glyph_if_exist:nTF { `#4 }
10161     { \__xunadd_add_circle:nN {#1} #4 }
10162     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
10163   }
10164 }
10165 \cs_new_protected:Npn \__xunadd_add_circle:nN #1#2
10166 {
10167   \hcoffin_set:Nn \l__xunadd_tmp_coffin {#1}
10168   \hcoffin_set:Nn \l__xunadd_circle_coffin {#2}
10169   \fp_set:Nn \l__xunadd_circle_scale_fp
10170   {
10171     \dim_to_decimal_in_unit:nn
10172     {
10173       \fp_use:N \l__xunadd_circle_ratio_fp
10174       \coffin_wd:N \l__xunadd_circle_coffin
10175     }
10176     { \coffin_wd:N \l__xunadd_tmp_coffin }
10177   }
10178   \coffin_scale:Nnn \l__xunadd_tmp_coffin
10179   { \l__xunadd_circle_scale_fp } { \l__xunadd_circle_scale_fp }
10180   \coffin_attach:NnnNnnnn
10181   \l__xunadd_circle_coffin { hc } { vc }
10182   \l__xunadd_tmp_coffin { hc } { vc } { \c_zero_dim } { \c_zero_dim }
10183   \coffin_typeset:Nnnnn \l__xunadd_circle_coffin
10184   { H } { l } { \c_zero_dim } { \c_zero_dim }
10185 }
10186 \fp_new:N \l__xunadd_circle_scale_fp
10187 \coffin_new:N \l__xunadd_tmp_coffin
10188 \coffin_new:N \l__xunadd_circle_coffin

```

设置圆圈中文字的宽度与圆圈宽度的比例, 预设为 0.7。

```

10189 \NewDocumentCommand \settextcircledratio { m }
10190 { \fp_set:Nn \l__xunadd_circle_ratio_fp {#1} }
10191 \fp_new:N \l__xunadd_circle_ratio_fp
10192 \settextcircledratio { 0.7 }

```

使 \t 等组合重音符号放在参数的第一个字母的右边。

```

10193 \cs_new_protected:Npn \__xunadd_combine_double_accent:nnNNn

```

```

10194 { \__xunadd_text_combine:NnnNNn \__xunadd_add_double_accent:nnNN }
10195 \cs_new_protected:Npn \__xunadd_add_double_accent:nnNN #1#2#3#4
10196 {
10197   \tl_if_blank:nTF {#1}
10198   {
10199     \__xunadd_glyph_if_exist:nTF { `#4 }
10200     {#4}
10201     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10202   }
10203   {
10204     \__xunadd_glyph_if_exist:nTF { `#3 }
10205     { \__xunadd_add_double_symbol:nN {#1} #3 }
10206     {
10207       \__xunadd_glyph_if_exist:nTF { `#4 }
10208       { \add@accent { `#4 } {#1} }
10209       { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10210     }
10211   }
10212 }

```

使 \sliding 等组合重音符号放在参数的第一个字母的右边。

```

10213 \cs_new_protected:Npn \__xunadd_combine_double_symbol:nnNNn
10214 { \__xunadd_text_combine:NnnNNn \__xunadd_add_double_symbol:nnNN }
10215 \cs_new_protected:Npn \__xunadd_add_double_symbol:nnNN #1#2#3#4
10216 {
10217   \tl_if_blank:nTF {#1}
10218   {
10219     \__xunadd_glyph_if_exist:nTF { `#4 }
10220     {#4}
10221     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
10222   }
10223   {
10224     \__xunadd_glyph_if_exist:nTF { `#3 }
10225     { \__xunadd_add_double_symbol:nN {#1} #3 }
10226     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
10227   }
10228 }

```

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。

```

10229 \cs_new_protected:Npn \__xunadd_add_double_symbol:nN #1#2
10230 {
10231   \tl_if_head_is_N_type:nTF {#1}
10232   {
10233     \exp_after:wN \exp_after:wN \exp_after:wN
10234     \__xunadd_add_double_symbol_aux:NnN \exp_after:wN \exp_after:wN
10235     \tl_head:w #1 \q_stop \exp_after:wN { \use_none:n #1 } #2
10236   }
10237   { #1#2 }
10238 }
10239 \cs_new_protected:Npn \__xunadd_add_double_symbol_aux:NnN #1#2#3
10240 {
10241   \bool_lazy_any:nTF
10242   {
10243     { \token_if_letter_p:N #1 }
10244     { \token_if_other_p:N #1 }
10245     { \token_if_chardef_p:N #1 }
10246   }
10247   { #1#3#2 }
10248   { #1#2#3 }
10249 }

```

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。

```

10250 \NewDocumentCommand \AtBeginUTFCommand { s O { } +m }

```

```

10251 {
10252   \tl_if_blank:nTF {#2}
10253   {
10254     \IfBooleanTF {#1}
10255     { \xunadd_set_begin_hook:n }
10256     { \xunadd_append_begin_hook:n }
10257   }
10258   { \xunadd_set_begin_hook:nn {#2} }
10259   {#3}
10260 }
10261 \NewDocumentCommand \AtEndUTFCommand { s O { } +m }
10262 {
10263   \tl_if_blank:nTF {#2}
10264   {
10265     \IfBooleanTF {#1}
10266     { \xunadd_set_end_hook:n }
10267     { \xunadd_append_end_hook:n }
10268   }
10269   { \xunadd_set_end_hook:nn {#2} }
10270   {#3}
10271 }

\xunadd_set_begin_hook:n 10272 \cs_new_protected:Npn \xunadd_set_begin_hook:n
\xunadd_set_end_hook:n 10273 { \tl_set:Nn \l__xunadd_begin_hook_tl }
10274 \cs_new_protected:Npn \xunadd_append_begin_hook:n
10275 { \tl_put_right:Nn \l__xunadd_begin_hook_tl }
10276 \cs_new_protected:Npn \xunadd_set_end_hook:n
10277 { \tl_set:Nn \l__xunadd_end_hook_tl }
10278 \cs_new_protected:Npn \xunadd_append_end_hook:n
10279 { \tl_put_right:Nn \l__xunadd_end_hook_tl }
10280 \cs_new_protected:Npn \xunadd_set_begin_hook:nn
10281 { \__xunadd_set_cmd_hook:nnn { begin } }
10282 \cs_new_protected:Npn \xunadd_set_end_hook:nn
10283 { \__xunadd_set_cmd_hook:nnn { end } }
10284 \cs_new_protected:Npn \__xunadd_set_cmd_hook:nnn #1#2#3
10285 {
10286   \cs_set_protected:cpn
10287   {
10288     \tl_if_single:nTF {#2}
10289     { \use:c { __xunadd_#1_csname:n } { \token_to_str:N #2 } }
10290     { \__xunadd_set_cmd_hook_aux:Nnwn #2 \q_stop {#1} }
10291   } ##1
10292   {#3}
10293 }
10294 \cs_new:Npn \__xunadd_set_cmd_hook_aux:Nnwn #1#2 \q_stop #3
10295 { \use:c { __xunadd_#3_csname:n } { \token_to_str:N #1 - \tl_to_str:n {#2} } }
10296 \cs_new:Npn \__xunadd_begin_csname:n #1 { __xunadd_begin_#1_hook:n }
10297 \cs_new:Npn \__xunadd_end_csname:n #1 { __xunadd_end_#1_hook:n }
10298 \tl_new:N \l__xunadd_begin_hook_tl
10299 \tl_new:N \l__xunadd_end_hook_tl

__xunadd_begin_hook:nn 10300 \cs_new_protected:Npn \__xunadd_begin_hook:nn #1#2
__xunadd_end_hook:nn 10301 {
10302   \tl_use:N \l__xunadd_begin_hook_tl
10303   \cs_if_exist_use:cF { \__xunadd_begin_csname:n { #1 - \tl_to_str:n {#2} } }
10304   { \cs_if_exist_use:cF { \__xunadd_begin_csname:n {#1} } { \use_none:n } }
10305   {#2}
10306 }
10307 \cs_new_protected:Npn \__xunadd_end_hook:nn #1#2
10308 {
10309   \cs_if_exist_use:cF { \__xunadd_end_csname:n { #1 - \tl_to_str:n {#2} } }
10310   { \cs_if_exist_use:cF { \__xunadd_end_csname:n {#1} } { \use_none:n } }
10311   {#2}
10312   \tl_use:N \l__xunadd_end_hook_tl
10313 }

\DeclareUTFTIPACCommand 10314 \NewDocumentCommand \DeclareUTFTIPACCommand { O { \UTFencname } m }
10315 { \use:e { \__xunadd_text_tipa_command:Nnn \exp_not:N #2 { \token_to_str:N #2 } {#1} } }

```

```

10316 \cs_new_protected:Npn \__xunadd_text_tipa_command:Nnn #1#2#3
10317 {
10318   \cs_set_eq:cc { UTF/#3#2 } { #3#2 }
10319   \DeclareTextCommand #1 {#3} { \__xunadd_text_tipa_command:nnn {#3} {#2} }
10320 }
10321 \cs_new_protected:Npn \__xunadd_text_tipa_command:nnn #1#2#3
10322 {
10323   \exp_args:Ncc \__xunadd_check_for_tipa:NNn
10324   { \use_none:n #2 } { UTF/#1#2 } {#3}
10325 }
10326 \cs_new_protected:Npn \__xunadd_check_for_tipa:NNn #1#2#3
10327 {
10328   \tl_if_head_eq_meaning:nNTF {#3} \textipa
10329   {
10330     \exp_after:wN \tipacatchonechar \exp_after:wN
10331     { \exp_after:wN #1 \use_none:n #3 }
10332   }
10333   { #2 {#3} }
10334 }

```

\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。

```

10335 \cs_new_protected:Npn \xunadd_get_slot:nn #1#2
10336 { \__xunadd_get_slot:wn #2 \q_nil \q_stop {#1} }
10337 \cs_new_protected:Npn \__xunadd_get_slot:wn #1#2#3 \q_stop #4
10338 {
10339   \int_set:Nn \l_xunadd_slot_int { -1 }
10340   \bool_set_false:N \l_xunadd_rest_bool
10341   \group_begin: \exp_args:Nccc \group_end:
10342   { \__xunadd_get_slot:NNnn }
10343   { #4 \token_to_str:N #1 }
10344   { \__xunadd_composite_cs:Nnn #1 {#4} {#2} }
10345   {#2}
10346   {#3}
10347 }
10348 \int_new:N \l_xunadd_slot_int
10349 \bool_new:N \l_xunadd_rest_bool
10350 \cs_new_protected:Npn \__xunadd_get_slot:NNnn #1#2#3#4
10351 {
10352   \cs_if_free:NF #1
10353   {
10354     \cs_if_exist:NTF #2
10355     { \__xunadd_get_composite_slot:Nn #2 {#4} }
10356     { \__xunadd_get_character_slot:Nn #1 { #3 #4 } }
10357   }
10358 }
10359 \cs_new_protected:Npn \__xunadd_get_composite_slot:Nn #1#2
10360 {
10361   \token_if_chardef:NT #1
10362   {
10363     \int_set:Nn \l_xunadd_slot_int {#1}
10364     \quark_if_nil:nF {#2}
10365     { \bool_set_true:N \l_xunadd_rest_bool }
10366   }
10367 }
10368 \cs_new_protected:Npn \__xunadd_get_character_slot:Nn #1
10369 {
10370   \exp_after:wN \__xunadd_get_character_slot_aux:wn #1
10371   \__xunadd_text_character:nN \q_nil \q_nil \q_stop
10372 }
10373 \cs_new_protected:Npn \__xunadd_get_character_slot_aux:wn
10374 #1 \__xunadd_text_character:nN #2#3#4 \q_stop #5
10375 {
10376   \quark_if_nil:nF {#2}
10377   {
10378     \int_set:Nn \l_xunadd_slot_int { `#3 }
10379     \quark_if_nil:nF {#5}
10380     { \bool_set_true:N \l_xunadd_rest_bool }

```

```

10381     }
10382 }

```

`\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。

```

10383 \cs_new_protected_nopar:Npn \xunadd@microtype@is@charx #1 \relax
10384 {
10385   \use:e
10386   { \xunadd_get_slot:nn { \MT@encoding } { \tex_the:D \MT@toks } }
10387   \int_compare:nNnTF \l_xunadd_slot_int < \c_zero_int
10388   { \xunadd@original@is@charx #1 \relax }
10389   {
10390     \cs_set_nopar:Npe \MT@char@ { \int_use:N \l_xunadd_slot_int }
10391     \bool_if:NT \l_xunadd_rest_bool { \MT@norestfalse }
10392   }
10393 }
10394 \cs_new_protected_nopar:Npn \xunadd@microtype@hook
10395 {
10396   \cs_if_free:NF \MT@is@charx
10397   {
10398     \cs_new_eq:NN \xunadd@original@is@charx \MT@is@charx
10399     \cs_set_eq:NN \MT@is@charx \xunadd@microtype@is@charx
10400     \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
10401   }
10402 }
10403 \ifpackageloaded { microtype }
10404 { \use:n } { \AtBeginDocument }
10405 { \xunadd@microtype@hook }

10406 </xunicode>
10407 <*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。

```

10408 \DeclareUTFSymbol\texthyphenationpoint{"2027}
10409 \DeclareUTFSymbol\texttwoemdash{"2E3A}

```

以下内容选自 `xunicode`, 并做了适当修改。

```

10410 \DeclareUTFComposite\textsuperscript
10411 \DeclareUTFComposite\textsubscript
10412 \DeclareUTFEncodedAccent\textsbleftarrow{"20EE}{02FF}
10413 \DeclareUTFEncodedAccent\`{"0300}{02CB}
10414 \DeclareUTFEncodedAccent\capitalgrave{"0300}{02CB}
10415 \DeclareUTFEncodedAccent\{'{"0301}{02CA}
10416 \DeclareUTFEncodedAccent\capitalacute{"0301}{02CA}
10417 \DeclareUTFEncodedAccent\~{"0302}{02C6}
10418 \DeclareUTFEncodedAccent\capitalcircumflex{"0302}{02C6}
10419 \DeclareUTFEncodedAccent\~{"0303}{02DC}
10420 \DeclareUTFEncodedAccent\capitaltilde{"0303}{02DC}
10421 \DeclareUTFEncodedAccent\={"0304}{02C9}
10422 \DeclareUTFEncodedAccent\capitalmacron{"0304}{02C9}
10423 \DeclareUTFEncodedAccent\textoverline{"0305}{020E}
10424 \DeclareUTFEncodedAccent\u{"0306}{02D8}
10425 \DeclareUTFEncodedAccent\capitalbreve{"0306}{02D8}
10426 \DeclareUTFEncodedAccent\.'{"0307}{02D9}
10427 \DeclareUTFEncodedAccent\capitaldotaccent{"0307}{02D9}
10428 \DeclareUTFEncodedAccent\{"0308}{00A8}
10429 \DeclareUTFEncodedAccent\capitaldieresis{"0308}{00A8}
10430 \DeclareUTFEncodedAccent\m{"0309}{0309}
10431 \DeclareUTFEncodedAccent\texthookabove{"0309}{0309}
10432 \DeclareUTFEncodedAccent\r{"030A}{02DA}
10433 \DeclareUTFEncodedAccent\capitalring{"030A}{02DA}
10434 \DeclareUTFEncodedAccent\H{"030B}{02DD}
10435 \DeclareUTFEncodedAccent\capitalhungarumlaut{"030B}{02DD}
10436 \DeclareUTFEncodedAccent\v{"030C}{02C7}

```

```

10437 \DeclareUTFEncodedAccent\capitalcaron{"030C}{ "02C7}
10438 \DeclareUTFEncodedAccent\textvbaraccent{"030D}{ "02C8}
10439 \DeclareUTFEncodedAccent\textdoublevbaraccent{"030E}{ "0022}
10440 \DeclareUTFEncodedAccent\U{"030E}{ "0022}
10441 \DeclareUTFEncodedAccent\textdoublegrave{"030F}{ "02F5}
10442 \DeclareUTFEncodedAccent\G{"030F}{ "02F5}
10443 \DeclareUTFEncodedAccent\textdotbreve{"0310}{ "0310}
10444 \DeclareUTFEncodedAccent\textroundcap{"0311}{ "0311}
10445 \DeclareUTFEncodedAccent\newtie{"0311}{ "0311}
10446 \DeclareUTFEncodedAccent\capitalnewtie{"0311}{ "0311}
10447 \DeclareUTFEncodedAccent\textturncommaabove{"0312}{ "02BB}
10448 \DeclareUTFEncodedAccent\textcommaabove{"0313}{ "02BC}
10449 \DeclareUTFEncodedAccent\textrevcommaabove{"0314}{ "02BD}
10450 \DeclareUTFEncodedAccent\overbridge{"0346}{ "0346}
10451 \DeclareUTFEncodedAccent\crtilde{"034A}{ "034A}
10452 \DeclareUTFEncodedAccent\dottedtilde{"034B}{ "034B}
10453 \DeclareUTFEncodedAccent\doubletilde{"034C}{ "034C}
10454 \DeclareUTFEncodedAccent\extrightarrowhead{"0350}{ "02C3}
10455 \DeclareUTFEncodedAccent\textlefthalfing{"0351}{ "02D3}
10456 \DeclareUTFEncodedAccent\extrighthalfing{"0357}{ "02D2}
10457 \DeclareUTFDoubleEncodedSymbol\textdoublebrevebelow{"035C}{ "035C}
10458 \DeclareUTFDoubleEncodedAccent\textdoublebreve{"035D}{ "035D}
10459 \DeclareUTFDoubleEncodedAccent\textdoublemacron{"035E}{ "035E}
10460 \DeclareUTFDoubleEncodedSymbol\textdoublemacronbelow{"035F}{ "035F}
10461 \DeclareUTFDoubleEncodedAccent\textdoubletilde{"0360}{ "0360}
10462 \DeclareUTFDoubleEncodedAccent\t{"0361}{ "0361}
10463 \DeclareUTFDoubleEncodedAccent\capitaltie{"0361}{ "0361}
10464 \DeclareUTFDoubleEncodedAccent\texttoptiebar{"0361}{ "0361}
10465 \DeclareUTFDoubleEncodedSymbol\sliding{"0362}{ "0362}
10466 \DeclareUTFTIPACCommand\t
10467 \DeclareUTFTIPACCommand\capitaltie
10468 \DeclareUTFTIPACCommand\texttoptiebar
10469 \DeclareUTFTIPACCommand\sliding
10470 \DeclareUTFEncodedAccent\texthighrise{"1DC4}{ "1DC4}
10471 \DeclareUTFEncodedAccent\textlowrise{"1DC5}{ "1DC5}
10472 \DeclareUTFEncodedAccent\textriseall{"1DC8}{ "1DC8}
10473 \DeclareUTFEncodedAccent\textfallrise{"1DC9}{ "1DC9}
10474 \DeclareUTFEncodedAccent\textaoilig{"1DD5}{ "1DD5}
10475 \DeclareUTFCompositeSymbol\textundertie{H}{ "1E2A}
10476 \DeclareUTFCompositeSymbol\textundertie{h}{ "1E2B}
10477 \DeclareUTFEncodedAccents\textcircumgrave{"0302}{ "0301}
10478 \DeclareUTFSymbol\textFinv{"2132}
10479 \DeclareUTFSymbol\textaleph{"2135}
10480 \DeclareUTFSymbol\textbeth{"2136}
10481 \DeclareUTFSymbol\textgimel{"2137}
10482 \DeclareUTFSymbol\textdaleth{"2138}
10483 \DeclareUTFSymbol\textGame{"2141}
10484 \DeclareUTFCompositeCommand\tonebar{25}{\tonebar{2}\tonebar{5}}
10485 \DeclareUTFCompositeCommand\tonebar{52}{\tonebar{5}\tonebar{2}}
10486 \DeclareUTFSymbol\textbigcircle{"25EF}
10487 \DeclareUTFEncodedCircle\textcircled{"20DD}{ "25EF}
10488 \DeclareUTFCompositeSymbol\textcircled{0}{ "24EA}
10489 \DeclareUTFCompositeSymbol\textcircled{1}{ "2460}
10490 \DeclareUTFCompositeSymbol\textcircled{2}{ "2461}
10491 \DeclareUTFCompositeSymbol\textcircled{3}{ "2462}
10492 \DeclareUTFCompositeSymbol\textcircled{4}{ "2463}
10493 \DeclareUTFCompositeSymbol\textcircled{5}{ "2464}
10494 \DeclareUTFCompositeSymbol\textcircled{6}{ "2465}
10495 \DeclareUTFCompositeSymbol\textcircled{7}{ "2466}
10496 \DeclareUTFCompositeSymbol\textcircled{8}{ "2467}
10497 \DeclareUTFCompositeSymbol\textcircled{9}{ "2468}
10498 \DeclareUTFCompositeSymbol\textcircled{10}{ "2469}
10499 \DeclareUTFCompositeSymbol\textcircled{11}{ "246A}
10500 \DeclareUTFCompositeSymbol\textcircled{12}{ "246B}
10501 \DeclareUTFCompositeSymbol\textcircled{13}{ "246C}
10502 \DeclareUTFCompositeSymbol\textcircled{14}{ "246D}
10503 \DeclareUTFCompositeSymbol\textcircled{15}{ "246E}

```

```
10504 \DeclareUTFCompositeSymbol\textcircled{16}{\textcircled{16}}{"246F}
10505 \DeclareUTFCompositeSymbol\textcircled{17}{\textcircled{17}}{"2470}
10506 \DeclareUTFCompositeSymbol\textcircled{18}{\textcircled{18}}{"2471}
10507 \DeclareUTFCompositeSymbol\textcircled{19}{\textcircled{19}}{"2472}
10508 \DeclareUTFCompositeSymbol\textcircled{20}{\textcircled{20}}{"2473}
10509 \DeclareUTFCompositeSymbol\textcircled{21}{\textcircled{21}}{"3251}
10510 \DeclareUTFCompositeSymbol\textcircled{22}{\textcircled{22}}{"3252}
10511 \DeclareUTFCompositeSymbol\textcircled{23}{\textcircled{23}}{"3253}
10512 \DeclareUTFCompositeSymbol\textcircled{24}{\textcircled{24}}{"3254}
10513 \DeclareUTFCompositeSymbol\textcircled{25}{\textcircled{25}}{"3255}
10514 \DeclareUTFCompositeSymbol\textcircled{26}{\textcircled{26}}{"3256}
10515 \DeclareUTFCompositeSymbol\textcircled{27}{\textcircled{27}}{"3257}
10516 \DeclareUTFCompositeSymbol\textcircled{28}{\textcircled{28}}{"3258}
10517 \DeclareUTFCompositeSymbol\textcircled{29}{\textcircled{29}}{"3259}
10518 \DeclareUTFCompositeSymbol\textcircled{30}{\textcircled{30}}{"325A}
10519 \DeclareUTFCompositeSymbol\textcircled{31}{\textcircled{31}}{"325B}
10520 \DeclareUTFCompositeSymbol\textcircled{32}{\textcircled{32}}{"325C}
10521 \DeclareUTFCompositeSymbol\textcircled{33}{\textcircled{33}}{"325D}
10522 \DeclareUTFCompositeSymbol\textcircled{34}{\textcircled{34}}{"325E}
10523 \DeclareUTFCompositeSymbol\textcircled{35}{\textcircled{35}}{"325F}
10524 \DeclareUTFCompositeSymbol\textcircled{36}{\textcircled{36}}{"32B1}
10525 \DeclareUTFCompositeSymbol\textcircled{37}{\textcircled{37}}{"32B2}
10526 \DeclareUTFCompositeSymbol\textcircled{38}{\textcircled{38}}{"32B3}
10527 \DeclareUTFCompositeSymbol\textcircled{39}{\textcircled{39}}{"32B4}
10528 \DeclareUTFCompositeSymbol\textcircled{40}{\textcircled{40}}{"32B5}
10529 \DeclareUTFCompositeSymbol\textcircled{41}{\textcircled{41}}{"32B6}
10530 \DeclareUTFCompositeSymbol\textcircled{42}{\textcircled{42}}{"32B7}
10531 \DeclareUTFCompositeSymbol\textcircled{43}{\textcircled{43}}{"32B8}
10532 \DeclareUTFCompositeSymbol\textcircled{44}{\textcircled{44}}{"32B9}
10533 \DeclareUTFCompositeSymbol\textcircled{45}{\textcircled{45}}{"32BA}
10534 \DeclareUTFCompositeSymbol\textcircled{46}{\textcircled{46}}{"32BB}
10535 \DeclareUTFCompositeSymbol\textcircled{47}{\textcircled{47}}{"32BC}
10536 \DeclareUTFCompositeSymbol\textcircled{48}{\textcircled{48}}{"32BD}
10537 \DeclareUTFCompositeSymbol\textcircled{49}{\textcircled{49}}{"32BE}
10538 \DeclareUTFCompositeSymbol\textcircled{50}{\textcircled{50}}{"32BF}
10539 \DeclareUTFCompositeSymbol\textcircled{A}{\textcircled{A}}{"24B6}
10540 \DeclareUTFCompositeSymbol\textcircled{B}{\textcircled{B}}{"24B7}
10541 \DeclareUTFCompositeSymbol\textcircled{C}{\textcircled{C}}{"24B8}
10542 \DeclareUTFCompositeSymbol\textcircled{D}{\textcircled{D}}{"24B9}
10543 \DeclareUTFCompositeSymbol\textcircled{E}{\textcircled{E}}{"24BA}
10544 \DeclareUTFCompositeSymbol\textcircled{F}{\textcircled{F}}{"24BB}
10545 \DeclareUTFCompositeSymbol\textcircled{G}{\textcircled{G}}{"24BC}
10546 \DeclareUTFCompositeSymbol\textcircled{H}{\textcircled{H}}{"24BD}
10547 \DeclareUTFCompositeSymbol\textcircled{I}{\textcircled{I}}{"24BE}
10548 \DeclareUTFCompositeSymbol\textcircled{J}{\textcircled{J}}{"24BF}
10549 \DeclareUTFCompositeSymbol\textcircled{K}{\textcircled{K}}{"24C0}
10550 \DeclareUTFCompositeSymbol\textcircled{L}{\textcircled{L}}{"24C1}
10551 \DeclareUTFCompositeSymbol\textcircled{M}{\textcircled{M}}{"24C2}
10552 \DeclareUTFCompositeSymbol\textcircled{N}{\textcircled{N}}{"24C3}
10553 \DeclareUTFCompositeSymbol\textcircled{O}{\textcircled{O}}{"24C4}
10554 \DeclareUTFCompositeSymbol\textcircled{P}{\textcircled{P}}{"24C5}
10555 \DeclareUTFCompositeSymbol\textcircled{Q}{\textcircled{Q}}{"24C6}
10556 \DeclareUTFCompositeSymbol\textcircled{R}{\textcircled{R}}{"24C7}
10557 \DeclareUTFCompositeSymbol\textcircled{S}{\textcircled{S}}{"24C8}
10558 \DeclareUTFCompositeSymbol\textcircled{T}{\textcircled{T}}{"24C9}
10559 \DeclareUTFCompositeSymbol\textcircled{U}{\textcircled{U}}{"24CA}
10560 \DeclareUTFCompositeSymbol\textcircled{V}{\textcircled{V}}{"24CB}
10561 \DeclareUTFCompositeSymbol\textcircled{W}{\textcircled{W}}{"24CC}
10562 \DeclareUTFCompositeSymbol\textcircled{X}{\textcircled{X}}{"24CD}
10563 \DeclareUTFCompositeSymbol\textcircled{Y}{\textcircled{Y}}{"24CE}
10564 \DeclareUTFCompositeSymbol\textcircled{Z}{\textcircled{Z}}{"24CF}
10565 \DeclareUTFCompositeSymbol\textcircled{a}{\textcircled{a}}{"24D0}
10566 \DeclareUTFCompositeSymbol\textcircled{b}{\textcircled{b}}{"24D1}
10567 \DeclareUTFCompositeSymbol\textcircled{c}{\textcircled{c}}{"24D2}
10568 \DeclareUTFCompositeSymbol\textcircled{d}{\textcircled{d}}{"24D3}
10569 \DeclareUTFCompositeSymbol\textcircled{e}{\textcircled{e}}{"24D4}
10570 \DeclareUTFCompositeSymbol\textcircled{f}{\textcircled{f}}{"24D5}
```

```

10571 \DeclareUTFCompositeSymbol\textcircled{g}{\textcircled{g}}{"24D6}
10572 \DeclareUTFCompositeSymbol\textcircled{h}{\textcircled{h}}{"24D7}
10573 \DeclareUTFCompositeSymbol\textcircled{i}{\textcircled{i}}{"24D8}
10574 \DeclareUTFCompositeSymbol\textcircled{j}{\textcircled{j}}{"24D9}
10575 \DeclareUTFCompositeSymbol\textcircled{k}{\textcircled{k}}{"24DA}
10576 \DeclareUTFCompositeSymbol\textcircled{l}{\textcircled{l}}{"24DB}
10577 \DeclareUTFCompositeSymbol\textcircled{m}{\textcircled{m}}{"24DC}
10578 \DeclareUTFCompositeSymbol\textcircled{n}{\textcircled{n}}{"24DD}
10579 \DeclareUTFCompositeSymbol\textcircled{o}{\textcircled{o}}{"24DE}
10580 \DeclareUTFCompositeSymbol\textcircled{p}{\textcircled{p}}{"24DF}
10581 \DeclareUTFCompositeSymbol\textcircled{q}{\textcircled{q}}{"24E0}
10582 \DeclareUTFCompositeSymbol\textcircled{r}{\textcircled{r}}{"24E1}
10583 \DeclareUTFCompositeSymbol\textcircled{s}{\textcircled{s}}{"24E2}
10584 \DeclareUTFCompositeSymbol\textcircled{t}{\textcircled{t}}{"24E3}
10585 \DeclareUTFCompositeSymbol\textcircled{u}{\textcircled{u}}{"24E4}
10586 \DeclareUTFCompositeSymbol\textcircled{v}{\textcircled{v}}{"24E5}
10587 \DeclareUTFCompositeSymbol\textcircled{w}{\textcircled{w}}{"24E6}
10588 \DeclareUTFCompositeSymbol\textcircled{x}{\textcircled{x}}{"24E7}
10589 \DeclareUTFCompositeSymbol\textcircled{y}{\textcircled{y}}{"24E8}
10590 \DeclareUTFCompositeSymbol\textcircled{z}{\textcircled{z}}{"24E9}
10591 \DeclareUTFCompositeSymbol\textsuperscript{h}{\textsuperscript{h}}{"02B0}
10592 \DeclareUTFCompositeSymbol\textsuperscript{\textth}{\textth}{"02B1}
10593 \DeclareUTFCompositeSymbol\textsuperscript{j}{\textsuperscript{j}}{"02B2}
10594 \DeclareUTFCompositeSymbol\textsuperscript{r}{\textsuperscript{r}}{"02B3}
10595 \DeclareUTFCompositeSymbol\textsuperscript{\textturnr}{\textturnr}{"02B4}
10596 \DeclareUTFCompositeSymbol\textsuperscript{\textturnrrtail}{\textturnrrtail}{"02B5}
10597 \DeclareUTFCompositeSymbol\textsuperscript{\textinvscr}{\textinvscr}{"02B6}
10598 \DeclareUTFCompositeSymbol\textsuperscript{w}{\textsuperscript{w}}{"02B7}
10599 \DeclareUTFCompositeSymbol\textsuperscript{y}{\textsuperscript{y}}{"02B8}
10600 \DeclareUTFCompositeSymbol\textsuperscript{\textbabygamma}{\textbabygamma}{"02E0}
10601 \DeclareUTFCompositeSymbol\textsuperscript{\textgammalatinsmall}{\textgammalatinsmall}{"02E0}
10602 \DeclareUTFCompositeSymbol\textsuperscript{l}{\textsuperscript{l}}{"02E1}
10603 \DeclareUTFCompositeSymbol\textsuperscript{s}{\textsuperscript{s}}{"02E2}
10604 \DeclareUTFCompositeSymbol\textsuperscript{x}{\textsuperscript{x}}{"02E3}
10605 \DeclareUTFCompositeSymbol\textsuperscript{\textrevglotstop}{\textrevglotstop}{"02E4}
10606 \DeclareUTFCompositeSymbol\textsuperscript{\textrevepsilon}{\textrevepsilon}{"1D4C}
10607 \DeclareUTFCompositeSymbol\textsuperscript{\cyrn}{\cyrn}{"1D78}
10608 \DeclareUTFCompositeSymbol\textsuperscript{\textbarsci}{\textbarsci}{"1DA7}
10609 \DeclareUTFCompositeSymbol\textsuperscript{V}{\textsuperscript{V}}{"2C7D}
10610 \DeclareUTFCompositeSymbol\textsuperscript{\textHbar}{\textHbar}{"A7F8}
10611 \DeclareUTFCompositeSymbol\textsuperscript{\textHslash}{\textHslash}{"A7F8}
10612 \DeclareUTFCompositeSymbol\textsuperscript{\oe}{\oe}{"A7F9}
10613 \DeclareUTFCompositeSymbol\textsubscript{h}{\textsubscript{h}}{"2095}
10614 \DeclareUTFCompositeSymbol\textsubscript{k}{\textsubscript{k}}{"2096}
10615 \DeclareUTFCompositeSymbol\textsubscript{l}{\textsubscript{l}}{"2097}
10616 \DeclareUTFCompositeSymbol\textsubscript{m}{\textsubscript{m}}{"2098}
10617 \DeclareUTFCompositeSymbol\textsubscript{n}{\textsubscript{n}}{"2099}
10618 \DeclareUTFCompositeSymbol\textsubscript{p}{\textsubscript{p}}{"209A}
10619 \DeclareUTFCompositeSymbol\textsubscript{s}{\textsubscript{s}}{"209B}
10620 \DeclareUTFCompositeSymbol\textsubscript{t}{\textsubscript{t}}{"209C}

```

以下定义取自 `hyperref` 的 `puenc.def`。

```

10621 \DeclareUTFEncodedAccent\textinvbreve{"0311}{\textinvbreve}{"0311}
10622 \DeclareUTFEncodedSymbol\textsubbreve{"032E}{\textsubbreve}{"032E}
10623 \DeclareUTFSymbol\textHT{"0009}
10624 \DeclareUTFSymbol\textLF{"000A}
10625 \DeclareUTFSymbol\textCR{"000D}
10626 \DeclareUTFSymbol\textnumbersign{"0023}
10627 \DeclareUTFSymbol\textparenleft{"0028}
10628 \DeclareUTFSymbol\textparenright{"0029}
10629 \DeclareUTFSymbol\textMVPlus{"002B}
10630 \DeclareUTFSymbol\textMVComma{"002C}
10631 \DeclareUTFSymbol\textMVMinus{"002D}
10632 \DeclareUTFSymbol\textMVPeriod{"002E}
10633 \DeclareUTFSymbol\textMVDivision{"002F}
10634 \DeclareUTFSymbol\textMVZero{"0030}
10635 \DeclareUTFSymbol\textMVOne{"0031}

```

```

10636 \DeclareUTFSymbol\textMVTwo{"0032}
10637 \DeclareUTFSymbol\textMVThree{"0033}
10638 \DeclareUTFSymbol\textMVFour{"0034}
10639 \DeclareUTFSymbol\textMVFive{"0035}
10640 \DeclareUTFSymbol\textMVSix{"0036}
10641 \DeclareUTFSymbol\textMVSeven{"0037}
10642 \DeclareUTFSymbol\textMVEight{"0038}
10643 \DeclareUTFSymbol\textMVNine{"0039}
10644 \DeclareUTFSymbol\textMVAt{"0040}
10645 \DeclareUTFCompositeCommand\.\{i}{i}
10646 \DeclareUTFCompositeCommand\.\{i}{i}
10647 \DeclareUTFSymbol\textlnot{"00AC}
10648 \DeclareUTFSymbol\textplusminus{"00B1}
10649 \DeclareUTFSymbol\textcedilla{"00B8}
10650 \DeclareUTFSymbol\textmultiply{"00D7}
10651 \DeclareUTFSymbol\textThorn{"00DE}
10652 \DeclareUTFSymbol\textdivide{"00F7}
10653 \DeclareUTFSymbol\textHslash{"0126}
10654 \DeclareUTFCompositeSymbol\k\{i}{{"012F}
10655 \DeclareUTFCompositeSymbol\.\{L}{{"013F}
10656 \DeclareUTFCompositeSymbol\.\{l}{{"0140}
10657 \DeclareUTFSymbol\textnapostrophe{"0149}
10658 \DeclareUTFSymbol\textTslash{"0166}
10659 \DeclareUTFSymbol\texttslash{"0167}
10660 \DeclareUTFSymbol\textlongs{"017F}
10661 \DeclareUTFSymbol\texthausA{"0181}
10662 \DeclareUTFSymbol\texthausD{"018A}
10663 \DeclareUTFSymbol\textrevE{"018E}
10664 \DeclareUTFSymbol\texthausK{"0198}
10665 \DeclareUTFSymbol\textPUnrleg{"019E}
10666 \DeclareUTFSymbol\textinve{"01DD}
10667 \DeclareUTFSymbol\textGslash{"01E4}
10668 \DeclareUTFSymbol\textgslash{"01E5}
10669 \DeclareUTFCompositeSymbol\textinvbreve{E}{{"0206}
10670 \DeclareUTFCompositeSymbol\textinvbreve{e}{{"0207}
10671 \DeclareUTFCompositeSymbol\textinvbreve{I}{{"020A}
10672 \DeclareUTFCompositeSymbol\textinvbreve{i}{{"020B}
10673 \DeclareUTFCompositeSymbol\textinvbreve{\i}{{"020B}
10674 \DeclareUTFCompositeSymbol\textinvbreve{O}{{"020E}
10675 \DeclareUTFCompositeSymbol\textinvbreve{o}{{"020F}
10676 \DeclareUTFCompositeSymbol\textinvbreve{U}{{"0216}
10677 \DeclareUTFCompositeSymbol\textinvbreve{u}{{"0217}
10678 \DeclareUTFSymbol\j{"0237}
10679 \DeclareUTFSymbol\textPUdblig{"0238}
10680 \DeclareUTFSymbol\textPUqplig{"0239}
10681 \DeclareUTFSymbol\textslashc{"023C}
10682 \DeclareUTFSymbol\textniepsilon{"025B}
10683 \DeclareUTFSymbol\textipagamma{"0263}
10684 \DeclareUTFSymbol\textniiota{"0269}
10685 \DeclareUTFSymbol\textnippi{"0278}
10686 \DeclareUTFSymbol\textniupsilon{"028A}
10687 \DeclareUTFSymbol\textring{"02DA}
10688 \DeclareUTFSymbol\texttilde{"02DC}
10689 \DeclareUTFSymbol\texthungarumlaut{"02DD}
10690 \DeclareUTFSymbol\textringlow{"02F3}
10691 \DeclareUTFSymbol\texttildelow{"02F7}
10692 \DeclareUTFCommand\textnewtie{\textinvbreve\ }
10693 \DeclareUTFCommand\textdotbelow{\d\ }
10694 \DeclareUTFSymbol\textmacronbelow{"02CD}
10695 \DeclareUTFCommand\texttie{\t\ }
10696 \DeclareUTFSymbol\textnumeralsigngreek{"0374}
10697 \DeclareUTFSymbol\textnumeralsignlowergreek{"0375}
10698 \DeclareUTFCompositeSymbol\'{\textAlpha}{{"0386}
10699 \DeclareUTFCompositeSymbol\'{\textEpsilon}{{"0388}
10700 \DeclareUTFCompositeSymbol\'{\textEta}{{"0389}
10701 \DeclareUTFCompositeSymbol\'{\textIota}{{"038A}
10702 \DeclareUTFCompositeSymbol\'{\textOmicron}{{"038C}

```

```

10703 \DeclareUTFCompositeSymbol\{'\textUpsilon>{"038E}
10704 \DeclareUTFCompositeSymbol\{'\textOmega>{"038F}
10705 \DeclareUTFCompositeSymbol\{'\textIotadieresis>{"0390}
10706 \DeclareUTFSymbol\textIotadieresis{"03AA}
10707 \DeclareUTFCompositeSymbol\{"\textIota>{"03AA}
10708 \DeclareUTFCompositeSymbol\{"\textUpsilon>{"03AB}
10709 \DeclareUTFCompositeSymbol\{'\textalpha>{"03AC}
10710 \DeclareUTFCompositeSymbol\{'\textepsilon>{"03AD}
10711 \DeclareUTFCompositeSymbol\{'\texteta>{"03AE}
10712 \DeclareUTFCompositeSymbol\{'\textiota>{"03AF}
10713 \DeclareUTFCompositeSymbol\{"\textupsilonacutet>{"03B0}
10714 \DeclareUTFSymbol\textmugreek{"03BC}
10715 \DeclareUTFSymbol\textvarsigma{"03C2}
10716 \DeclareUTFCompositeSymbol\{"\textiota>{"03CA}
10717 \DeclareUTFCompositeSymbol\{"\textupsilon>{"03CB}
10718 \DeclareUTFCompositeSymbol\{'\textomicron>{"03CC}
10719 \DeclareUTFSymbol\textupsilonacutet{"03CD}
10720 \DeclareUTFCompositeSymbol\{'\textupsilon>{"03CD}
10721 \DeclareUTFCompositeSymbol\{'\textomega>{"03CE}
10722 \DeclareUTFSymbol\textStigmagreek{"03DA}
10723 \DeclareUTFSymbol\textstigmagreek{"03DB}
10724 \DeclareUTFSymbol\textDigammagreek{"03DC}
10725 \DeclareUTFSymbol\textdigammagreek{"03DD}
10726 \DeclareUTFSymbol\textKoppagreek{"03DE}
10727 \DeclareUTFSymbol\textkoppagreek{"03DF}
10728 \DeclareUTFSymbol\textSampigreek{"03E0}
10729 \DeclareUTFSymbol\textsampigreek{"03E1}
10730 \DeclareUTFSymbol\textbackepsilon{"03F6}
10731 \DeclareUTFCompositeSymbol\{'\CYRE>{"0400}
10732 \DeclareUTFSymbol\CYRYO{"0401}
10733 \DeclareUTFCompositeSymbol\{'\CYRE>{"0401}
10734 \DeclareUTFSymbol\CYRDJE{"0402}
10735 \DeclareUTFCompositeSymbol\{'\CYRG>{"0403}
10736 \DeclareUTFSymbol\CYRIE{"0404}
10737 \DeclareUTFSymbol\CYRDZE{"0405}
10738 \DeclareUTFSymbol\CYRII{"0406}
10739 \DeclareUTFSymbol\CYRYI{"0407}
10740 \DeclareUTFCompositeSymbol\{'\CYRII>{"0407}
10741 \DeclareUTFSymbol\CYRJE{"0408}
10742 \DeclareUTFSymbol\CYRLJE{"0409}
10743 \DeclareUTFSymbol\CYRNJE{"040A}
10744 \DeclareUTFSymbol\CYRTSHE{"040B}
10745 \DeclareUTFCompositeSymbol\{'\CYRK>{"040C}
10746 \DeclareUTFCompositeSymbol\{'\CYRI>{"040D}
10747 \DeclareUTFSymbol\CYRUSHRT{"040E}
10748 \DeclareUTFCompositeSymbol\U{"\CYRU>{"040E}
10749 \DeclareUTFSymbol\CYRDZHE{"040F}
10750 \DeclareUTFSymbol\CYRA{"0410}
10751 \DeclareUTFSymbol\CYRB{"0411}
10752 \DeclareUTFSymbol\CYRV{"0412}
10753 \DeclareUTFSymbol\CYRG{"0413}
10754 \DeclareUTFSymbol\CYRD{"0414}
10755 \DeclareUTFSymbol\CYRE{"0415}
10756 \DeclareUTFSymbol\CYRZH{"0416}
10757 \DeclareUTFSymbol\CYRZ{"0417}
10758 \DeclareUTFSymbol\CYRI{"0418}
10759 \DeclareUTFSymbol\CYRISHRT{"0419}
10760 \DeclareUTFCompositeSymbol\U{"\CYRI>{"0419}
10761 \DeclareUTFSymbol\CYRK{"041A}
10762 \DeclareUTFSymbol\CYRL{"041B}
10763 \DeclareUTFSymbol\CYRM{"041C}
10764 \DeclareUTFSymbol\CYRN{"041D}
10765 \DeclareUTFSymbol\CYRO{"041E}
10766 \DeclareUTFSymbol\CYRP{"041F}
10767 \DeclareUTFSymbol\CYRR{"0420}
10768 \DeclareUTFSymbol\CYRS{"0421}
10769 \DeclareUTFSymbol\CYRT{"0422}

```

```

10770 \DeclareUTFSymbol\CYRU{"0423}
10771 \DeclareUTFSymbol\CYRF{"0424}
10772 \DeclareUTFSymbol\CYRH{"0425}
10773 \DeclareUTFSymbol\CYRC{"0426}
10774 \DeclareUTFSymbol\CYRCH{"0427}
10775 \DeclareUTFSymbol\CYRSH{"0428}
10776 \DeclareUTFSymbol\CYRSHCH{"0429}
10777 \DeclareUTFSymbol\CYRHRDSN{"042A}
10778 \DeclareUTFSymbol\CYRERY{"042B}
10779 \DeclareUTFSymbol\CYRSFTSN{"042C}
10780 \DeclareUTFSymbol\CYREREV{"042D}
10781 \DeclareUTFSymbol\CYRYU{"042E}
10782 \DeclareUTFSymbol\CYRYA{"042F}
10783 \DeclareUTFSymbol\cyra{"0430}
10784 \DeclareUTFSymbol\cyrb{"0431}
10785 \DeclareUTFSymbol\cyrv{"0432}
10786 \DeclareUTFSymbol\cyrg{"0433}
10787 \DeclareUTFSymbol\cyrd{"0434}
10788 \DeclareUTFSymbol\cyre{"0435}
10789 \DeclareUTFSymbol\cyrrzh{"0436}
10790 \DeclareUTFSymbol\cyrrz{"0437}
10791 \DeclareUTFSymbol\cyri{"0438}
10792 \DeclareUTFSymbol\cyrishrt{"0439}
10793 \DeclareUTFCompositeSymbol\U{\cyri}{0439}
10794 \DeclareUTFSymbol\cyrk{"043A}
10795 \DeclareUTFSymbol\cyr1{"043B}
10796 \DeclareUTFSymbol\cyrn{"043C}
10797 \DeclareUTFSymbol\cyrn{"043D}
10798 \DeclareUTFSymbol\cyro{"043E}
10799 \DeclareUTFSymbol\cyrp{"043F}
10800 \DeclareUTFSymbol\cyrr{"0440}
10801 \DeclareUTFSymbol\cyrs{"0441}
10802 \DeclareUTFSymbol\cyrt{"0442}
10803 \DeclareUTFSymbol\cyru{"0443}
10804 \DeclareUTFSymbol\cyrf{"0444}
10805 \DeclareUTFSymbol\cyrh{"0445}
10806 \DeclareUTFSymbol\cyrc{"0446}
10807 \DeclareUTFSymbol\cyrch{"0447}
10808 \DeclareUTFSymbol\cyrsh{"0448}
10809 \DeclareUTFSymbol\cyrshch{"0449}
10810 \DeclareUTFSymbol\cyrhrdsn{"044A}
10811 \DeclareUTFSymbol\cyrery{"044B}
10812 \DeclareUTFSymbol\cyrsftsn{"044C}
10813 \DeclareUTFSymbol\cyrerev{"044D}
10814 \DeclareUTFSymbol\cyryu{"044E}
10815 \DeclareUTFSymbol\cyrya{"044F}
10816 \DeclareUTFCompositeSymbol\{\cyre}{0450}
10817 \DeclareUTFSymbol\cyryo{"0451}
10818 \DeclareUTFCompositeSymbol\{\cyre}{0451}
10819 \DeclareUTFSymbol\cyrdje{"0452}
10820 \DeclareUTFCompositeSymbol\'\cyrg{"0453}
10821 \DeclareUTFSymbol\cyrie{"0454}
10822 \DeclareUTFSymbol\cyrdze{"0455}
10823 \DeclareUTFSymbol\cyrii{"0456}
10824 \DeclareUTFSymbol\cyryi{"0457}
10825 \DeclareUTFCompositeSymbol\{\cyrii}{0457}
10826 \DeclareUTFSymbol\cyrje{"0458}
10827 \DeclareUTFSymbol\cyr1je{"0459}
10828 \DeclareUTFSymbol\cyrnje{"045A}
10829 \DeclareUTFSymbol\cyrtshe{"045B}
10830 \DeclareUTFCompositeSymbol\'\cyrk{"045C}
10831 \DeclareUTFCompositeSymbol\{\cyri}{045D}
10832 \DeclareUTFSymbol\cyrushrt{"045E}
10833 \DeclareUTFCompositeSymbol\U{\cyru}{045E}
10834 \DeclareUTFSymbol\cyrdzhe{"045F}
10835 \DeclareUTFSymbol\CYROMEGA{"0460}
10836 \DeclareUTFSymbol\cyromega{"0461}

```

```

10837 \DeclareUTFSymbol\CYRYAT{"0462}
10838 \DeclareUTFSymbol\cyryat{"0463}
10839 \DeclareUTFSymbol\CYRIOTE{"0464}
10840 \DeclareUTFSymbol\cyriote{"0465}
10841 \DeclareUTFSymbol\CYRLYUS{"0466}
10842 \DeclareUTFSymbol\cyrlyus{"0467}
10843 \DeclareUTFSymbol\CYRIOTLYUS{"0468}
10844 \DeclareUTFSymbol\cyriotlyus{"0469}
10845 \DeclareUTFSymbol\CYRBYUS{"046A}
10846 \DeclareUTFSymbol\cyrbyus{"046B}
10847 \DeclareUTFSymbol\CYRIOTBYUS{"046C}
10848 \DeclareUTFSymbol\cyriotbyus{"046D}
10849 \DeclareUTFSymbol\CYRKSI{"046E}
10850 \DeclareUTFSymbol\cyrksi{"046F}
10851 \DeclareUTFSymbol\CYRPSI{"0470}
10852 \DeclareUTFSymbol\cyrpsi{"0471}
10853 \DeclareUTFSymbol\CYRFITA{"0472}
10854 \DeclareUTFSymbol\cyrfita{"0473}
10855 \DeclareUTFSymbol\CYRIZH{"0474}
10856 \DeclareUTFSymbol\cyrizh{"0475}
10857 \DeclareUTFCompositeSymbol\C{\CYRIZH}{0476}
10858 \DeclareUTFCompositeSymbol\C{\cyrizh}{0477}
10859 \DeclareUTFSymbol\CYRUK{"0478}
10860 \DeclareUTFSymbol\cyruk{"0479}
10861 \DeclareUTFSymbol\CYROMEGARND{"047A}
10862 \DeclareUTFSymbol\cyromegarnd{"047B}
10863 \DeclareUTFSymbol\CYROMEGATITLO{"047C}
10864 \DeclareUTFSymbol\cyromegatitlo{"047D}
10865 \DeclareUTFSymbol\CYROT{"047E}
10866 \DeclareUTFSymbol\cyrot{"047F}
10867 \DeclareUTFSymbol\CYRKOPPA{"0480}
10868 \DeclareUTFSymbol\cyrkoppa{"0481}
10869 \DeclareUTFSymbol\cyrthousands{"0482}
10870 \DeclareUTFSymbol\CYRISHRTDSC{"048A}
10871 \DeclareUTFSymbol\cyrishrtdsc{"048B}
10872 \DeclareUTFSymbol\CYRSEMISFTSN{"048C}
10873 \DeclareUTFSymbol\cyrsemisftsn{"048D}
10874 \DeclareUTFSymbol\CYRRTICK{"048E}
10875 \DeclareUTFSymbol\cyrrtick{"048F}
10876 \DeclareUTFSymbol\CYRGUP{"0490}
10877 \DeclareUTFSymbol\cyrgup{"0491}
10878 \DeclareUTFSymbol\CYRGHCRS{"0492}
10879 \DeclareUTFSymbol\cyrghcrs{"0493}
10880 \DeclareUTFSymbol\CYRGHK{"0494}
10881 \DeclareUTFSymbol\cyrghk{"0495}
10882 \DeclareUTFSymbol\CYRZHDSC{"0496}
10883 \DeclareUTFSymbol\cyrzhdsc{"0497}
10884 \DeclareUTFSymbol\CYRZDSC{"0498}
10885 \DeclareUTFCompositeSymbol\c{\CYRZ}{0498}
10886 \DeclareUTFSymbol\cyrzdsc{"0499}
10887 \DeclareUTFCompositeSymbol\c{\cyrz}{0499}
10888 \DeclareUTFSymbol\CYRKDSC{"049A}
10889 \DeclareUTFSymbol\cyrkdsc{"049B}
10890 \DeclareUTFSymbol\CYRKVCRS{"049C}
10891 \DeclareUTFSymbol\cyrkvcrs{"049D}
10892 \DeclareUTFSymbol\CYRKHCRS{"049E}
10893 \DeclareUTFSymbol\cyrkhcrs{"049F}
10894 \DeclareUTFSymbol\CYRKBEAK{"04A0}
10895 \DeclareUTFSymbol\cyrkbeak{"04A1}
10896 \DeclareUTFSymbol\CYRNDSC{"04A2}
10897 \DeclareUTFSymbol\cyrndsc{"04A3}
10898 \DeclareUTFSymbol\CYRNG{"04A4}
10899 \DeclareUTFSymbol\cyrng{"04A5}
10900 \DeclareUTFSymbol\CYRPHK{"04A6}
10901 \DeclareUTFSymbol\cyrphk{"04A7}
10902 \DeclareUTFSymbol\CYRABHHA{"04A8}
10903 \DeclareUTFSymbol\cyrabhha{"04A9}

```

```

10904 \DeclareUTFSymbol\CYRSDSC{"04AA}
10905 \DeclareUTFCompositeSymbol\k{\CYRS}{04AA}
10906 \DeclareUTFSymbol\cyrsdsc{"04AB}
10907 \DeclareUTFCompositeSymbol\k{\cyrs}{04AB}
10908 \DeclareUTFSymbol\CYRTDSC{"04AC}
10909 \DeclareUTFSymbol\cyrtddsc{"04AD}
10910 \DeclareUTFSymbol\CYRY{"04AE}
10911 \DeclareUTFSymbol\cyry{"04AF}
10912 \DeclareUTFSymbol\CYRYHCRS{"04B0}
10913 \DeclareUTFSymbol\cyryhcrs{"04B1}
10914 \DeclareUTFSymbol\CYRHDSC{"04B2}
10915 \DeclareUTFSymbol\cyrhddsc{"04B3}
10916 \DeclareUTFSymbol\CYRTETSE{"04B4}
10917 \DeclareUTFSymbol\cyrtetse{"04B5}
10918 \DeclareUTFSymbol\CYRCHRDSC{"04B6}
10919 \DeclareUTFSymbol\cyrchrdsc{"04B7}
10920 \DeclareUTFSymbol\CYRCHVCRS{"04B8}
10921 \DeclareUTFSymbol\cyrchvcrs{"04B9}
10922 \DeclareUTFSymbol\CYRSHHA{"04BA}
10923 \DeclareUTFSymbol\cyrshha{"04BB}
10924 \DeclareUTFSymbol\CYRABHCH{"04BC}
10925 \DeclareUTFSymbol\cyrabhch{"04BD}
10926 \DeclareUTFSymbol\CYRABHCHDSC{"04BE}
10927 \DeclareUTFCompositeSymbol\k{\CYRABHCH}{04BE}
10928 \DeclareUTFSymbol\cyrabhchddsc{"04BF}
10929 \DeclareUTFCompositeSymbol\k{\cyrabhch}{04BF}
10930 \DeclareUTFSymbol\CYRpalochka{"04C0}
10931 \DeclareUTFCompositeSymbol\U{\CYRZH}{04C1}
10932 \DeclareUTFCompositeSymbol\U{\cyrz}{04C2}
10933 \DeclareUTFSymbol\CYRKHK{"04C3}
10934 \DeclareUTFSymbol\cyrkhh{"04C4}
10935 \DeclareUTFSymbol\CYRLDSC{"04C5}
10936 \DeclareUTFSymbol\cyrlddsc{"04C6}
10937 \DeclareUTFSymbol\CYRNHK{"04C7}
10938 \DeclareUTFSymbol\cyrnhh{"04C8}
10939 \DeclareUTFSymbol\CYRCHLDSC{"04CB}
10940 \DeclareUTFSymbol\cyrchldsc{"04CC}
10941 \DeclareUTFSymbol\CYRMDSC{"04CD}
10942 \DeclareUTFSymbol\cyrmdsc{"04CE}
10943 \DeclareUTFCompositeSymbol\U{\CYRA}{04D0}
10944 \DeclareUTFCompositeSymbol\U{\cyra}{04D1}
10945 \DeclareUTFCompositeSymbol\{"{\CYRA}{04D2}
10946 \DeclareUTFCompositeSymbol\{"{\cyra}{04D3}
10947 \DeclareUTFSymbol\CYRAE{"04D4}
10948 \DeclareUTFSymbol\cyrae{"04D5}
10949 \DeclareUTFCompositeSymbol\U{\CYRE}{04D6}
10950 \DeclareUTFCompositeSymbol\U{\cyre}{04D7}
10951 \DeclareUTFSymbol\CYRSCHWA{"04D8}
10952 \DeclareUTFSymbol\cyrschwa{"04D9}
10953 \DeclareUTFCompositeSymbol\{"{\CYRSCHWA}{04DA}
10954 \DeclareUTFCompositeSymbol\{"{\cyrschwa}{04DB}
10955 \DeclareUTFCompositeSymbol\{"{\CYRZH}{04DC}
10956 \DeclareUTFCompositeSymbol\{"{\cyrz}{04DD}
10957 \DeclareUTFCompositeSymbol\{"{\CYRZ}{04DE}
10958 \DeclareUTFCompositeSymbol\{"{\cyrz}{04DF}
10959 \DeclareUTFSymbol\CYRABHDZE{"04E0}
10960 \DeclareUTFSymbol\cyrabhdze{"04E1}
10961 \DeclareUTFCompositeSymbol\={"{\CYRI}{04E2}
10962 \DeclareUTFCompositeSymbol\={"{\cyri}{04E3}
10963 \DeclareUTFCompositeSymbol\{"{\CYRI}{04E4}
10964 \DeclareUTFCompositeSymbol\{"{\cyri}{04E5}
10965 \DeclareUTFCompositeSymbol\{"{\CYRO}{04E6}
10966 \DeclareUTFCompositeSymbol\{"{\cyro}{04E7}
10967 \DeclareUTFSymbol\CYROTLD{"04E8}
10968 \DeclareUTFSymbol\cyrotld{"04E9}
10969 \DeclareUTFCompositeSymbol\{"{\CYROTLD}{04EA}
10970 \DeclareUTFCompositeSymbol\{"{\cyrotld}{04EB}

```

```

10971 \DeclareUTFCompositeSymbol\{"\CYREREV}{04EC}
10972 \DeclareUTFCompositeSymbol\{"\cyrerev}{04ED}
10973 \DeclareUTFCompositeSymbol\={"\CYRU}{04EE}
10974 \DeclareUTFCompositeSymbol\={"\cyru}{04EF}
10975 \DeclareUTFCompositeSymbol\{"\CYRU}{04F0}
10976 \DeclareUTFCompositeSymbol\{"\cyru}{04F1}
10977 \DeclareUTFCompositeSymbol\H{\CYRU}{04F2}
10978 \DeclareUTFCompositeSymbol\H{\cyru}{04F3}
10979 \DeclareUTFCompositeSymbol\{"\CYRCH}{04F4}
10980 \DeclareUTFCompositeSymbol\{"\cyrch}{04F5}
10981 \DeclareUTFSymbol\CYRGDSC{04F6}
10982 \DeclareUTFSymbol\cyrgdsc{04F7}
10983 \DeclareUTFCompositeSymbol\{"\CYRERY}{04F8}
10984 \DeclareUTFCompositeSymbol\{"\cyrery}{04F9}
10985 \DeclareUTFSymbol\CYRHHK{04FC}
10986 \DeclareUTFSymbol\cyrhkh{04FD}
10987 \DeclareUTFSymbol\sofpasuq{05C3}
10988 \DeclareUTFSymbol\hebralef{05D0}
10989 \DeclareUTFSymbol\hebbet{05D1}
10990 \DeclareUTFSymbol\hebgimel{05D2}
10991 \DeclareUTFSymbol\hebdalet{05D3}
10992 \DeclareUTFSymbol\hebhe{05D4}
10993 \DeclareUTFSymbol\hebvav{05D5}
10994 \DeclareUTFSymbol\hebzayin{05D6}
10995 \DeclareUTFSymbol\hebbet{05D7}
10996 \DeclareUTFSymbol\hebtet{05D8}
10997 \DeclareUTFSymbol\hebyod{05D9}
10998 \DeclareUTFSymbol\hebfinalkaf{05DA}
10999 \DeclareUTFSymbol\hebkaf{05DB}
11000 \DeclareUTFSymbol\heblamed{05DC}
11001 \DeclareUTFSymbol\hebfinalmem{05DD}
11002 \DeclareUTFSymbol\hebmam{05DE}
11003 \DeclareUTFSymbol\hebfinalnun{05DF}
11004 \DeclareUTFSymbol\hebnun{05E0}
11005 \DeclareUTFSymbol\hebsamekh{05E1}
11006 \DeclareUTFSymbol\hebayin{05E2}
11007 \DeclareUTFSymbol\hebfinalpe{05E3}
11008 \DeclareUTFSymbol\hebpe{05E4}
11009 \DeclareUTFSymbol\hebfinaltsadi{05E5}
11010 \DeclareUTFSymbol\hebtsadi{05E6}
11011 \DeclareUTFSymbol\hebqof{05E7}
11012 \DeclareUTFSymbol\hebreph{05E8}
11013 \DeclareUTFSymbol\hebshin{05E9}
11014 \DeclareUTFSymbol\hebtav{05EA}
11015 \DeclareUTFSymbol\doublevav{05F0}
11016 \DeclareUTFSymbol\vavyod{05F1}
11017 \DeclareUTFSymbol\doubleyod{05F2}
11018 \DeclareUTFSymbol\textscd{1D05}
11019 \DeclareUTFSymbol\textPUsck{1D0B}
11020 \DeclareUTFSymbol\textPUscm{1D0D}
11021 \DeclareUTFSymbol\textPUscp{1D18}
11022 \DeclareUTFSymbol\textPUrevscr{1D19}
11023 \DeclareUTFSymbol\textiinferior{1D62}
11024 \DeclareUTFSymbol\textrinferior{1D63}
11025 \DeclareUTFSymbol\textuinferior{1D64}
11026 \DeclareUTFSymbol\textvinferior{1D65}
11027 \DeclareUTFSymbol\textbetainferior{1D66}
11028 \DeclareUTFSymbol\textgammainferior{1D67}
11029 \DeclareUTFSymbol\textrhoinferior{1D68}
11030 \DeclareUTFSymbol\textphiinferior{1D69}
11031 \DeclareUTFSymbol\textchiinferior{1D6A}
11032 \DeclareUTFSymbol\textbarsci{1D7B}
11033 \DeclareUTFSymbol\textbarp{1D7D}
11034 \DeclareUTFSymbol\textbarscu{1D7E}
11035 \DeclareUTFSymbol\textPurhooka{1D8F}
11036 \DeclareUTFSymbol\textPurhooke{1D92}
11037 \DeclareUTFSymbol\textPurhookepsilon{1D93}

```

```

11038 \DeclareUTFSymbol\textPurhookopeno{"1D97}
11039 \DeclareUTFCompositeSymbol\textsubbreve{H}{{"1E2A}
11040 \DeclareUTFCompositeSymbol\textsubbreve{h}{{"1E2B}
11041 \DeclareUTFCompositeSymbol\.\{\textlongS}{{"1E9B}
11042 \DeclareUTFSymbol\textcompwordmark{"200C}
11043 \DeclareUTFSymbol\texthdotfor{"2025}
11044 \DeclareUTFSymbol\textprime{"2032}
11045 \DeclareUTFSymbol\textsecond{"2033}
11046 \DeclareUTFSymbol\textthird{"2034}
11047 \DeclareUTFSymbol\textbackprime{"2035}
11048 \DeclareUTFSymbol\textlefttherefore{"2056}
11049 \DeclareUTFSymbol\textfourth{"2057}
11050 \DeclareUTFSymbol\textdiamonddots{"2058}
11051 \DeclareUTFSymbol\textzerosuperior{"2070}
11052 \DeclareUTFSymbol\textisuperior{"2071}
11053 \DeclareUTFSymbol\textfoursuperior{"2074}
11054 \DeclareUTFSymbol\textfivesuperior{"2075}
11055 \DeclareUTFSymbol\textsixsuperior{"2076}
11056 \DeclareUTFSymbol\textsevensuperior{"2077}
11057 \DeclareUTFSymbol\texteightsuperior{"2078}
11058 \DeclareUTFSymbol\textninesuperior{"2079}
11059 \DeclareUTFSymbol\textplussuperior{"207A}
11060 \DeclareUTFSymbol\textminussuperior{"207B}
11061 \DeclareUTFSymbol\textequalssuperior{"207C}
11062 \DeclareUTFSymbol\textparenleftsuperior{"207D}
11063 \DeclareUTFSymbol\textparenrightsuperior{"207E}
11064 \DeclareUTFSymbol\textnsuperior{"207F}
11065 \DeclareUTFSymbol\textzeroinferior{"2080}
11066 \DeclareUTFSymbol\textoneinferior{"2081}
11067 \DeclareUTFSymbol\texttwoinferior{"2082}
11068 \DeclareUTFSymbol\textthreeinferior{"2083}
11069 \DeclareUTFSymbol\textfourinferior{"2084}
11070 \DeclareUTFSymbol\textfiveinferior{"2085}
11071 \DeclareUTFSymbol\textsixinferior{"2086}
11072 \DeclareUTFSymbol\textseveninferior{"2087}
11073 \DeclareUTFSymbol\texteightinferior{"2088}
11074 \DeclareUTFSymbol\textnineinferior{"2089}
11075 \DeclareUTFSymbol\textplusinferior{"208A}
11076 \DeclareUTFSymbol\textminusinferior{"208B}
11077 \DeclareUTFSymbol\textequalsinferior{"208C}
11078 \DeclareUTFSymbol\textparenleftinferior{"208D}
11079 \DeclareUTFSymbol\textparenrightinferior{"208E}
11080 \DeclareUTFSymbol\textainferior{"2090}
11081 \DeclareUTFSymbol\texteinferior{"2091}
11082 \DeclareUTFSymbol\textoinferior{"2092}
11083 \DeclareUTFSymbol\textxinferior{"2093}
11084 \DeclareUTFSymbol\textschwainferior{"2094}
11085 \DeclareUTFSymbol\texthinferior{"2095}
11086 \DeclareUTFSymbol\textkinferior{"2096}
11087 \DeclareUTFSymbol\textlinferior{"2097}
11088 \DeclareUTFSymbol\textminferior{"2098}
11089 \DeclareUTFSymbol\textninferior{"2099}
11090 \DeclareUTFSymbol\textpinferior{"209A}
11091 \DeclareUTFSymbol\textsinferior{"209B}
11092 \DeclareUTFSymbol\texttinferior{"209C}
11093 \DeclareUTFSymbol\textpeseta{"20A7}
11094 \DeclareUTFSymbol\textDeleatur{"20B0}
11095 \DeclareUTFSymbol\textguarani{"20B2}
11096 \DeclareUTFSymbol\textslash{"210F}
11097 \DeclareUTFSymbol\textIm{"2111}
11098 \DeclareUTFSymbol\textell{"2113}
11099 \DeclareUTFSymbol\textwp{"2118}
11100 \DeclareUTFSymbol\textRe{"211C}
11101 \DeclareUTFSymbol\textriota{"2129}
11102 \DeclareUTFSymbol\textangstrom{"212B}
11103 \DeclareUTFSymbol\textfax{"213B}
11104 \DeclareUTFSymbol\textinvamp{"214B}

```

```

11105 \DeclareUTFSymbol\textoneseventh{"2150}
11106 \DeclareUTFSymbol\textoneninth{"2151}
11107 \DeclareUTFSymbol\textonetenth{"2152}
11108 \DeclareUTFSymbol\textonethird{"2153}
11109 \DeclareUTFSymbol\texttwothirds{"2154}
11110 \DeclareUTFSymbol\textonefifth{"2155}
11111 \DeclareUTFSymbol\texttwofifths{"2156}
11112 \DeclareUTFSymbol\textthreefifths{"2157}
11113 \DeclareUTFSymbol\textfourfifths{"2158}
11114 \DeclareUTFSymbol\textonesixth{"2159}
11115 \DeclareUTFSymbol\textfivesixths{"215A}
11116 \DeclareUTFSymbol\textoneeighth{"215B}
11117 \DeclareUTFSymbol\textthreeeighths{"215C}
11118 \DeclareUTFSymbol\textfiveeighths{"215D}
11119 \DeclareUTFSymbol\textseveeneighths{"215E}
11120 \DeclareUTFSymbol\textrevc{"2184}
11121 \DeclareUTFSymbol\textzerothirds{"2189}
11122 \DeclareUTFSymbol\textnleftarrow{"219A}
11123 \DeclareUTFSymbol\textnrightarrow{"219B}
11124 \DeclareUTFSymbol\texttwoheadleftarrow{"219E}
11125 \DeclareUTFCommand\textntwoheadleftarrow{\textlstrikethru\texttwoheadleftarrow}
11126 \DeclareUTFSymbol\texttwoheaduparrow{"219F}
11127 \DeclareUTFSymbol\texttwoheadrightarrow{"21A0}
11128 \DeclareUTFCommand\textntwoheadrightarrow{\textlstrikethru\texttwoheadrightarrow}
11129 \DeclareUTFSymbol\texttwoheaddownarrow{"21A1}
11130 \DeclareUTFSymbol\textleftarrowtail{"21A2}
11131 \DeclareUTFSymbol\textrightarrowtail{"21A3}
11132 \DeclareUTFSymbol\textmapsto{"21A6}
11133 \DeclareUTFSymbol\texthookleftarrow{"21A9}
11134 \DeclareUTFSymbol\texthookrightarrow{"21AA}
11135 \DeclareUTFSymbol\textlooparrowleft{"21AB}
11136 \DeclareUTFSymbol\textlooparrowright{"21AC}
11137 \DeclareUTFSymbol\textnleftrightarrow{"21AE}
11138 \DeclareUTFSymbol\textlightning{"21AF}
11139 \DeclareUTFSymbol\textdls{"21B5}
11140 \DeclareUTFSymbol\textcurvearrowleft{"21B6}
11141 \DeclareUTFSymbol\textcurvearrowright{"21B7}
11142 \DeclareUTFSymbol\textleftharpoonup{"21BC}
11143 \DeclareUTFSymbol\textleftharpoondown{"21BD}
11144 \DeclareUTFSymbol\textupharpoonright{"21BE}
11145 \DeclareUTFSymbol\textupharpoonleft{"21BF}
11146 \DeclareUTFSymbol\textrightharpoonup{"21C0}
11147 \DeclareUTFSymbol\textrightharpoondown{"21C1}
11148 \DeclareUTFSymbol\textdownharpoonright{"21C2}
11149 \DeclareUTFSymbol\textdownharpoonleft{"21C3}
11150 \DeclareUTFSymbol\textrightleftarrows{"21C4}
11151 \DeclareUTFSymbol\textupdownarrows{"21C5}
11152 \DeclareUTFSymbol\textleftrightrightarrows{"21C6}
11153 \DeclareUTFSymbol\textleftleftarrows{"21C7}
11154 \DeclareUTFSymbol\textupuparrows{"21C8}
11155 \DeclareUTFSymbol\textrighttrightarrows{"21C9}
11156 \DeclareUTFSymbol\textdowndownarrows{"21CA}
11157 \DeclareUTFSymbol\textleftrightharpoons{"21CB}
11158 \DeclareUTFSymbol\textrightleftharpoons{"21CC}
11159 \DeclareUTFSymbol\textnLeftarrow{"21CD}
11160 \DeclareUTFSymbol\textnLeftrightarrow{"21CE}
11161 \DeclareUTFSymbol\textnRightarrow{"21CF}
11162 \DeclareUTFSymbol\textLeftarrow{"21D0}
11163 \DeclareUTFSymbol\textUparrow{"21D1}
11164 \DeclareUTFSymbol\textRightarrow{"21D2}
11165 \DeclareUTFSymbol\textDownarrow{"21D3}
11166 \DeclareUTFSymbol\textLeftrightarrow{"21D4}
11167 \DeclareUTFSymbol\textUpdownarrow{"21D5}
11168 \DeclareUTFSymbol\textNarrow{"21D6}
11169 \DeclareUTFSymbol\textNearrow{"21D7}
11170 \DeclareUTFSymbol\textSearrow{"21D8}
11171 \DeclareUTFSymbol\textSwarrow{"21D9}

```

```

11172 \DeclareUTFSymbol\textLleftarrow{"21DA}
11173 \DeclareUTFSymbol\textRrightarrow{"21DB}
11174 \DeclareUTFSymbol\textleftsquigarrow{"21DC}
11175 \DeclareUTFSymbol\textright squigarrow{"21DD}
11176 \DeclareUTFSymbol\textdashleftarrow{"21E0}
11177 \DeclareUTFSymbol\textdasheduparrow{"21E1}
11178 \DeclareUTFSymbol\textdashrightarrow{"21E2}
11179 \DeclareUTFSymbol\textdasheddownarrow{"21E3}
11180 \DeclareUTFSymbol\textpointer{"21E8}
11181 \DeclareUTFSymbol\textdownuparrows{"21F5}
11182 \DeclareUTFSymbol\textleftarrowtriangle{"21FD}
11183 \DeclareUTFSymbol\textrightarrowtriangle{"21FE}
11184 \DeclareUTFSymbol\textleftrightharpoontriangle{"21FF}
11185 \DeclareUTFSymbol\textforall{"2200}
11186 \DeclareUTFSymbol\textcomplement{"2201}
11187 \DeclareUTFSymbol\textpartial{"2202}
11188 \DeclareUTFSymbol\textexists{"2203}
11189 \DeclareUTFSymbol\textnexists{"2204}
11190 \DeclareUTFSymbol\textemptyset{"2205}
11191 \DeclareUTFSymbol\texttriangle{"2206}
11192 \DeclareUTFSymbol\textnabla{"2207}
11193 \DeclareUTFSymbol\textin{"2208}
11194 \DeclareUTFSymbol\textnotin{"2209}
11195 \DeclareUTFSymbol\textsmallin{"220A}
11196 \DeclareUTFSymbol\textni{"220B}
11197 \DeclareUTFSymbol\textnotowner{"220C}
11198 \DeclareUTFSymbol\textsmallowns{"220D}
11199 \DeclareUTFSymbol\textprod{"220F}
11200 \DeclareUTFSymbol\textamalg{"2210}
11201 \DeclareUTFSymbol\textsum{"2211}
11202 \DeclareUTFSymbol\textmp{"2213}
11203 \DeclareUTFSymbol\textdotplus{"2214}
11204 \DeclareUTFSymbol\textDivides{"2215}
11205 \DeclareUTFSymbol\textsetminus{"2216}
11206 \DeclareUTFSymbol\textast{"2217}
11207 \DeclareUTFSymbol\textcirc{"2218}
11208 \DeclareUTFSymbol\textbulletoperator{"2219}
11209 \DeclareUTFSymbol\textpropto{"221D}
11210 \DeclareUTFSymbol\textinfty{"221E}
11211 \DeclareUTFSymbol\textangle{"2220}
11212 \DeclareUTFSymbol\textmeasuredangle{"2221}
11213 \DeclareUTFSymbol\textsphericalangle{"2222}
11214 \DeclareUTFSymbol\textmid{"2223}
11215 \DeclareUTFSymbol\textnmid{"2224}
11216 \DeclareUTFSymbol\textparallel{"2225}
11217 \DeclareUTFSymbol\textnparallel{"2226}
11218 \DeclareUTFSymbol\textwedge{"2227}
11219 \DeclareUTFCommand\textowedge{\textcircled\textwedge}
11220 \DeclareUTFSymbol\textvee{"2228}
11221 \DeclareUTFCommand\textovee{\textcircled\textvee}
11222 \DeclareUTFSymbol\textcap{"2229}
11223 \DeclareUTFSymbol\textcup{"222A}
11224 \DeclareUTFSymbol\textint{"222B}
11225 \DeclareUTFSymbol\textiint{"222C}
11226 \DeclareUTFSymbol\textiiint{"222D}
11227 \DeclareUTFSymbol\textoint{"222E}
11228 \DeclareUTFSymbol\textoiint{"222F}
11229 \DeclareUTFSymbol\textointclockwise{"2232}
11230 \DeclareUTFSymbol\textointctrclockwise{"2233}
11231 \DeclareUTFSymbol\texttherefore{"2234}
11232 \DeclareUTFSymbol\textbecause{"2235}
11233 \DeclareUTFSymbol\textvdotdot{"2236}
11234 \DeclareUTFSymbol\textsquaredots{"2237}
11235 \DeclareUTFSymbol\textdotminus{"2238}
11236 \DeclareUTFSymbol\texteqcolon{"2239}
11237 \DeclareUTFSymbol\textsim{"223C}
11238 \DeclareUTFSymbol\textbacksim{"223D}

```

```

11239 \DeclareUTFCommand\textnbacksim{\textlstrikethru\textbacksim}
11240 \DeclareUTFSymbol\textwr{"2240}
11241 \DeclareUTFSymbol\textnsim{"2241}
11242 \DeclareUTFSymbol\texteqsim{"2242}
11243 \DeclareUTFCommand\textneqsim{\textlstrikethru\texteqsim}
11244 \DeclareUTFSymbol\textsimeq{"2243}
11245 \DeclareUTFSymbol\textnsimeq{"2244}
11246 \DeclareUTFSymbol\textcong{"2245}
11247 \DeclareUTFSymbol\textncong{"2247}
11248 \DeclareUTFSymbol\textapprox{"2248}
11249 \DeclareUTFSymbol\textnapprox{"2249}
11250 \DeclareUTFSymbol\textapproxeq{"224A}
11251 \DeclareUTFCommand\textnapproxeq{\textlstrikethru\textapproxeq}
11252 \DeclareUTFSymbol\texttriplesim{"224B}
11253 \DeclareUTFCommand\textntriplesim{\textlstrikethru\texttriplesim}
11254 \DeclareUTFSymbol\textbackcong{"224C}
11255 \DeclareUTFCommand\textnbackcong{\textlstrikethru\textbackcong}
11256 \DeclareUTFSymbol\textasympt{"224D}
11257 \DeclareUTFCommand\textnasympt{\textlstrikethru\textasympt}
11258 \DeclareUTFSymbol\textBumpeq{"224E}
11259 \DeclareUTFCommand\textnBumpeq{\textlstrikethru\textBumpeq}
11260 \DeclareUTFSymbol\textbumpeq{"224F}
11261 \DeclareUTFCommand\textnbumpeq{\textlstrikethru\textbumpeq}
11262 \DeclareUTFSymbol\textdoteq{"2250}
11263 \DeclareUTFCommand\textndoteq{\textlstrikethru\textdoteq}
11264 \DeclareUTFSymbol\textdoteqdot{"2251}
11265 \DeclareUTFCommand\textnDoteq{\textlstrikethru\textdoteqdot}
11266 \DeclareUTFSymbol\textfallingdoteq{"2252}
11267 \DeclareUTFCommand\textnfallingdoteq{\textlstrikethru\textfallingdoteq}
11268 \DeclareUTFSymbol\textrisingdoteq{"2253}
11269 \DeclareUTFCommand\textnrisingdoteq{\textlstrikethru\textrisingdoteq}
11270 \DeclareUTFSymbol\textcolonequals{"2254}
11271 \DeclareUTFSymbol\textequalscolon{"2255}
11272 \DeclareUTFSymbol\texteqcirc{"2256}
11273 \DeclareUTFCommand\textneqcirc{\textlstrikethru\texteqcirc}
11274 \DeclareUTFSymbol\textcirceq{"2257}
11275 \DeclareUTFCommand\textncirceq{\textlstrikethru\textcirceq}
11276 \DeclareUTFSymbol\texthateq{"2259}
11277 \DeclareUTFCommand\textnhateq{\textlstrikethru\texthateq}
11278 \DeclareUTFSymbol\texttriangleeq{"225C}
11279 \DeclareUTFSymbol\textneq{"2260}
11280 \DeclareUTFSymbol\textne{"2260}
11281 \DeclareUTFSymbol\textequiv{"2261}
11282 \DeclareUTFSymbol\textnequiv{"2262}
11283 \DeclareUTFSymbol\textleq{"2264}
11284 \DeclareUTFSymbol\textle{"2264}
11285 \DeclareUTFSymbol\textgeq{"2265}
11286 \DeclareUTFSymbol\textge{"2265}
11287 \DeclareUTFSymbol\textleqq{"2266}
11288 \DeclareUTFCommand\textnleqq{\textlstrikethru\textleqq}
11289 \DeclareUTFSymbol\textgeqq{"2267}
11290 \DeclareUTFCommand\textngeqq{\textlstrikethru\textgeqq}
11291 \DeclareUTFSymbol\textlneqq{"2268}
11292 \DeclareUTFSymbol\textgneqq{"2269}
11293 \DeclareUTFSymbol\textll{"226A}
11294 \DeclareUTFCommand\textnll{\textlstrikethru\textll}
11295 \DeclareUTFSymbol\textgg{"226B}
11296 \DeclareUTFCommand\textngg{\textlstrikethru\textgg}
11297 \DeclareUTFSymbol\textbetween{"226C}
11298 \DeclareUTFSymbol\textnless{"226E}
11299 \DeclareUTFSymbol\textngtr{"226F}
11300 \DeclareUTFSymbol\textnleq{"2270}
11301 \DeclareUTFSymbol\textngeq{"2271}
11302 \DeclareUTFSymbol\textlessssim{"2272}
11303 \DeclareUTFSymbol\textgtrsim{"2273}
11304 \DeclareUTFSymbol\textnlesssim{"2274}
11305 \DeclareUTFSymbol\textngtrsim{"2275}

```

```

11306 \DeclareUTFSymbol\textlessgtr{"2276}
11307 \DeclareUTFSymbol\textgtrless{"2277}
11308 \DeclareUTFSymbol\textngtrless{"2278}
11309 \DeclareUTFSymbol\textnlessgtr{"2279}
11310 \DeclareUTFSymbol\textprec{"227A}
11311 \DeclareUTFSymbol\textsucc{"227B}
11312 \DeclareUTFSymbol\textpreccurlyeq{"227C}
11313 \DeclareUTFSymbol\textsucccurlyeq{"227D}
11314 \DeclareUTFSymbol\textprecsim{"227E}
11315 \DeclareUTFCommand\textnprecsim{\textlstrikethru\textprecsim}
11316 \DeclareUTFSymbol\textsuccsim{"227F}
11317 \DeclareUTFCommand\textnsuccsim{\textlstrikethru\textsuccsim}
11318 \DeclareUTFSymbol\textnprec{"2280}
11319 \DeclareUTFSymbol\textnsucc{"2281}
11320 \DeclareUTFSymbol\textsubset{"2282}
11321 \DeclareUTFSymbol\textsupset{"2283}
11322 \DeclareUTFSymbol\textnsubset{"2284}
11323 \DeclareUTFSymbol\textnsupset{"2285}
11324 \DeclareUTFSymbol\textsubseteq{"2286}
11325 \DeclareUTFSymbol\textsupseteq{"2287}
11326 \DeclareUTFSymbol\textnsubseteq{"2288}
11327 \DeclareUTFSymbol\textnsupseteq{"2289}
11328 \DeclareUTFSymbol\textsubsetneq{"228A}
11329 \DeclareUTFSymbol\textsupsetneq{"228B}
11330 \DeclareUTFSymbol\textcupdot{"228D}
11331 \DeclareUTFSymbol\textcupplus{"228E}
11332 \DeclareUTFSymbol\textsqsubset{"228F}
11333 \DeclareUTFCommand\textnsqsubset{\textlstrikethru\textsqsubset}
11334 \DeclareUTFSymbol\textsqsupset{"2290}
11335 \DeclareUTFCommand\textnsqsupset{\textlstrikethru\textsqsupset}
11336 \DeclareUTFSymbol\textsqsubseq{"2291}
11337 \DeclareUTFCommand\textnsqsubseq{\textlstrikethru\textsqsubseq}
11338 \DeclareUTFSymbol\textsqsupseq{"2292}
11339 \DeclareUTFCommand\textnsqsupseq{\textlstrikethru\textsqsupseq}
11340 \DeclareUTFSymbol\textsqcap{"2293}
11341 \DeclareUTFSymbol\textsqcup{"2294}
11342 \DeclareUTFSymbol\textplus{"2295}
11343 \DeclareUTFSymbol\textominus{"2296}
11344 \DeclareUTFSymbol\textotimes{"2297}
11345 \DeclareUTFSymbol\textoslash{"2298}
11346 \DeclareUTFSymbol\textodot{"2299}
11347 \DeclareUTFSymbol\textcircledcirc{"229A}
11348 \DeclareUTFSymbol\textcircledast{"229B}
11349 \DeclareUTFSymbol\textcircledash{"229D}
11350 \DeclareUTFSymbol\textboxplus{"229E}
11351 \DeclareUTFSymbol\textboxminus{"229F}
11352 \DeclareUTFSymbol\textboxtimes{"22A0}
11353 \DeclareUTFSymbol\textboxdot{"22A1}
11354 \DeclareUTFSymbol\textvdash{"22A2}
11355 \DeclareUTFSymbol\textdashv{"22A3}
11356 \DeclareUTFCommand\textndashv{\textlstrikethru\textdashv}
11357 \DeclareUTFSymbol\texttop{"22A4}
11358 \DeclareUTFCommand\textndownvdash{\textlstrikethru\texttop}
11359 \DeclareUTFSymbol\textbot{"22A5}
11360 \DeclareUTFCommand\textnupvdash{\textlstrikethru\textbot}
11361 \DeclareUTFSymbol\textvDash{"22A8}
11362 \DeclareUTFSymbol\textVdash{"22A9}
11363 \DeclareUTFSymbol\textVvdash{"22AA}
11364 \DeclareUTFCommand\textnVdash{\textlstrikethru\textVvdash}
11365 \DeclareUTFSymbol\textVDash{"22AB}
11366 \DeclareUTFSymbol\textnvdash{"22AC}
11367 \DeclareUTFSymbol\textnvDash{"22AD}
11368 \DeclareUTFSymbol\textnVdash{"22AE}
11369 \DeclareUTFSymbol\textnVDash{"22AF}
11370 \DeclareUTFSymbol\textlhd{"22B2}
11371 \DeclareUTFSymbol\textrhd{"22B3}
11372 \DeclareUTFSymbol\textunlhd{"22B4}

```

```

11373 \DeclareUTFSymbol\textunrhd{"22B5}
11374 \DeclareUTFSymbol\textmultimapdotbothA{"22B6}
11375 \DeclareUTFSymbol\textmultimapdotbothB{"22B7}
11376 \DeclareUTFSymbol\textmultimap{"22B8}
11377 \DeclareUTFSymbol\textveebar{"22BB}
11378 \DeclareUTFSymbol\textbarwedgedge{"22BC}
11379 \DeclareUTFSymbol\textstar{"22C6}
11380 \DeclareUTFSymbol\textdivideontimes{"22C7}
11381 \DeclareUTFSymbol\textbowtie{"22C8}
11382 \DeclareUTFSymbol\textltimes{"22C9}
11383 \DeclareUTFSymbol\textrtimes{"22CA}
11384 \DeclareUTFSymbol\textleftthreetimes{"22CB}
11385 \DeclareUTFSymbol\textrightthreetimes{"22CC}
11386 \DeclareUTFSymbol\textbacksimeq{"22CD}
11387 \DeclareUTFCommand\textnbacksimeq{\textlstrikethru\textbacksimeq}
11388 \DeclareUTFSymbol\textcurlyvee{"22CE}
11389 \DeclareUTFSymbol\textcurlywedge{"22CF}
11390 \DeclareUTFSymbol\textSubset{"22D0}
11391 \DeclareUTFCommand\textnSubset{\textlstrikethru\textSubset}
11392 \DeclareUTFSymbol\textSupset{"22D1}
11393 \DeclareUTFCommand\textnSupset{\textlstrikethru\textSupset}
11394 \DeclareUTFSymbol\textCap{"22D2}
11395 \DeclareUTFSymbol\textCup{"22D3}
11396 \DeclareUTFSymbol\textpitchfork{"22D4}
11397 \DeclareUTFSymbol\textlessdot{"22D6}
11398 \DeclareUTFSymbol\textgtrdot{"22D7}
11399 \DeclareUTFSymbol\textlll{"22D8}
11400 \DeclareUTFSymbol\textggg{"22D9}
11401 \DeclareUTFSymbol\textlesseqgtr{"22DA}
11402 \DeclareUTFSymbol\textgtreqless{"22DB}
11403 \DeclareUTFSymbol\textcurlyeqprec{"22DE}
11404 \DeclareUTFCommand\textncurlyeqprec{\textlstrikethru\textcurlyeqprec}
11405 \DeclareUTFSymbol\textcurlyeqsucc{"22DF}
11406 \DeclareUTFCommand\textncurlyeqsucc{\textlstrikethru\textcurlyeqsucc}
11407 \DeclareUTFSymbol\textnpreccurlyeq{"22E0}
11408 \DeclareUTFSymbol\textnsucccurlyeq{"22E1}
11409 \DeclareUTFSymbol\textnqsubsepeq{"22E2}
11410 \DeclareUTFSymbol\textnqsupsepeq{"22E3}
11411 \DeclareUTFSymbol\textsqsubsepeq{"22E4}
11412 \DeclareUTFSymbol\textsqsupsepeq{"22E5}
11413 \DeclareUTFSymbol\textlnsim{"22E6}
11414 \DeclareUTFSymbol\textgnsim{"22E7}
11415 \DeclareUTFSymbol\textprecnsim{"22E8}
11416 \DeclareUTFSymbol\textsuccnsim{"22E9}
11417 \DeclareUTFSymbol\textntriangleleft{"22EA}
11418 \DeclareUTFSymbol\textntriangleright{"22EB}
11419 \DeclareUTFSymbol\textntrianglelefteq{"22EC}
11420 \DeclareUTFSymbol\textntrianglerighteq{"22ED}
11421 \DeclareUTFSymbol\textvdots{"22EE}
11422 \DeclareUTFSymbol\textcdots{"22EF}
11423 \DeclareUTFSymbol\textudots{"22F0}
11424 \DeclareUTFSymbol\textddots{"22F1}
11425 \DeclareUTFSymbol\textbarin{"22F6}
11426 \DeclareUTFSymbol\textdiameter{"2300}
11427 \DeclareUTFSymbol\textbackneg{"2310}
11428 \DeclareUTFSymbol\textwasylozenge{"2311}
11429 \DeclareUTFSymbol\textinvbackneg{"2319}
11430 \DeclareUTFSymbol\textclock{"231A}
11431 \DeclareUTFSymbol\textulcorner{"231C}
11432 \DeclareUTFSymbol\texturcorner{"231D}
11433 \DeclareUTFSymbol\textllcorner{"231E}
11434 \DeclareUTFSymbol\textlrcorner{"231F}
11435 \DeclareUTFSymbol\textfrown{"2322}
11436 \DeclareUTFSymbol\textsmile{"2323}
11437 \DeclareUTFSymbol\textKeyboard{"2328}
11438 \DeclareUTFSymbol\textlangle{"2329}
11439 \DeclareUTFSymbol\textrangle{"232A}

```

```

11440 \DeclareUTFSymbol\textAPLin{\textAPLin{"2339}}
11441 \DeclareUTFSymbol\textTumbler{"233C}
11442 \DeclareUTFSymbol\textstmaryrdbaro{"233D}
11443 \DeclareUTFSymbol\textnotslash{"233F}
11444 \DeclareUTFSymbol\textnotbackslash{"2340}
11445 \DeclareUTFSymbol\textboxbackslash{"2342}
11446 \DeclareUTFSymbol\textAPLleftarrowbox{"2347}
11447 \DeclareUTFSymbol\textAPLrightarrowbox{"2348}
11448 \DeclareUTFSymbol\textAPLuparrowbox{"2350}
11449 \DeclareUTFSymbol\textAPLdownarrowbox{"2357}
11450 \DeclareUTFSymbol\textAPLinput{"235E}
11451 \DeclareUTFSymbol\textRequest{"2370}
11452 \DeclareUTFSymbol\textBeam{"2393}
11453 \DeclareUTFSymbol\texthexagon{"2394}
11454 \DeclareUTFSymbol\textAPLbox{"2395}
11455 \DeclareUTFSymbol\textForwardToIndex{"23ED}
11456 \DeclareUTFSymbol\textRewindToIndex{"23EE}
11457 \DeclareUTFSymbol\textbbslash{"244A}
11458 \DeclareUTFSymbol\textCircledA{"24B6}
11459 \DeclareUTFSymbol\textCleaningF{"24BB}
11460 \DeclareUTFCommand\textCleaningFF{\b\textCleaningF}
11461 \DeclareUTFSymbol\textCleaningP{"24C5}
11462 \DeclareUTFCommand\textCleaningPP{\b\textCleaningP}
11463 \DeclareUTFSymbol\textCuttingLine{"2504}
11464 \DeclareUTFSymbol\textUParrow{"25B2}
11465 \DeclareUTFSymbol\textbigtriangleup{"25B3}
11466 \DeclareUTFSymbol\textForward{"25B6}
11467 \DeclareUTFSymbol\texttriangleright{"25B7}
11468 \DeclareUTFSymbol\textRHD{"25BA}
11469 \DeclareUTFSymbol\textDOWNarrow{"25BC}
11470 \DeclareUTFSymbol\textbigtriangledown{"25BD}
11471 \DeclareUTFSymbol\textRewind{"25C0}
11472 \DeclareUTFSymbol\texttriangleleft{"25C1}
11473 \DeclareUTFSymbol\textLHD{"25C4}
11474 \DeclareUTFSymbol\textdiamond{"25C7}
11475 \DeclareUTFSymbol\textlozenge{"25CA}
11476 \DeclareUTFSymbol\textLEFTCIRCLE{"25D6}
11477 \DeclareUTFSymbol\textRIGHTCIRCLE{"25D7}
11478 \DeclareUTFSymbol\textboxbar{"25EB}
11479 \DeclareUTFSymbol\textCloud{"2601}
11480 \DeclareUTFSymbol\textFiveStar{"2605}
11481 \DeclareUTFSymbol\textFiveStarOpen{"2606}
11482 \DeclareUTFSymbol\textPhone{"260E}
11483 \DeclareUTFSymbol\textboxempty{"2610}
11484 \DeclareUTFSymbol\textCheckedbox{"2611}
11485 \DeclareUTFSymbol\textCrossedbox{"2612}
11486 \DeclareUTFSymbol\textCoffeecup{"2615}
11487 \DeclareUTFSymbol\textHandCuffLeft{"261A}
11488 \DeclareUTFSymbol\textHandCuffRight{"261B}
11489 \DeclareUTFSymbol\textHandLeft{"261C}
11490 \DeclareUTFSymbol\textHandRight{"261E}
11491 \DeclareUTFSymbol\textRadioactivity{"2622}
11492 \DeclareUTFSymbol\textBiohazard{"2623}
11493 \DeclareUTFSymbol\textAnkh{"2625}
11494 \DeclareUTFSymbol\textYinYang{"262F}
11495 \DeclareUTFSymbol\textfrownie{"2639}
11496 \DeclareUTFSymbol\textsmiley{"263A}
11497 \DeclareUTFSymbol\textblacksmiley{"263B}
11498 \DeclareUTFSymbol\textsun{"263C}
11499 \DeclareUTFSymbol\textleftmoon{"263D}
11500 \DeclareUTFSymbol\textrightmoon{"263E}
11501 \DeclareUTFSymbol\textmercury{"263F}
11502 \DeclareUTFSymbol\textPUfemale{"2640}
11503 \DeclareUTFSymbol\textearth{"2641}
11504 \DeclareUTFSymbol\textmale{"2642}
11505 \DeclareUTFSymbol\textjupiter{"2643}
11506 \DeclareUTFSymbol\textsaturn{"2644}

```

```
11507 \DeclareUTFSymbol\texturanus{"2645}
11508 \DeclareUTFSymbol\textneptune{"2646}
11509 \DeclareUTFSymbol\textpluto{"2647}
11510 \DeclareUTFSymbol\textaries{"2648}
11511 \DeclareUTFSymbol\texttaurus{"2649}
11512 \DeclareUTFSymbol\textgemini{"264A}
11513 \DeclareUTFSymbol\textcancer{"264B}
11514 \DeclareUTFSymbol\textleo{"264C}
11515 \DeclareUTFSymbol\textvirgo{"264D}
11516 \DeclareUTFSymbol\textlibra{"264E}
11517 \DeclareUTFSymbol\textscorpio{"264F}
11518 \DeclareUTFSymbol\textsagittarius{"2650}
11519 \DeclareUTFSymbol\textcapricornus{"2651}
11520 \DeclareUTFSymbol\textaquarius{"2652}
11521 \DeclareUTFSymbol\textpisces{"2653}
11522 \DeclareUTFSymbol\textspadesuitblack{"2660}
11523 \DeclareUTFSymbol\textheartsuitwhite{"2661}
11524 \DeclareUTFSymbol\textdiamondsuitwhite{"2662}
11525 \DeclareUTFSymbol\textclubsuitblack{"2663}
11526 \DeclareUTFSymbol\textspadesuitwhite{"2664}
11527 \DeclareUTFSymbol\textheartsuitblack{"2665}
11528 \DeclareUTFSymbol\textdiamondsuitblack{"2666}
11529 \DeclareUTFSymbol\textclubsuitwhite{"2667}
11530 \DeclareUTFSymbol\textquaternote{"2669}
11531 \DeclareUTFSymbol\texttwoonotes{"266B}
11532 \DeclareUTFSymbol\textsixteenthnote{"266C}
11533 \DeclareUTFSymbol\textflat{"266D}
11534 \DeclareUTFSymbol\textnatural{"266E}
11535 \DeclareUTFSymbol\textsharp{"266F}
11536 \DeclareUTFSymbol\textrecycle{"2672}
11537 \DeclareUTFSymbol\textWheelchair{"267F}
11538 \DeclareUTFSymbol\textFlag{"2691}
11539 \DeclareUTFSymbol\textMineSign{"2692}
11540 \DeclareUTFSymbol\textdsimilitary{"2694}
11541 \DeclareUTFSymbol\textdsmedical{"2695}
11542 \DeclareUTFSymbol\textdsjuridical{"2696}
11543 \DeclareUTFSymbol\textdschemical{"2697}
11544 \DeclareUTFSymbol\textdsbiological{"2698}
11545 \DeclareUTFSymbol\textdscommercial{"269A}
11546 \DeclareUTFSymbol\textmanstar{"269D}
11547 \DeclareUTFSymbol\textdanger{"26A0}
11548 \DeclareUTFSymbol\textFemaleFemale{"26A2}
11549 \DeclareUTFSymbol\textMaleMale{"26A3}
11550 \DeclareUTFSymbol\textFemaleMale{"26A4}
11551 \DeclareUTFSymbol\textHermaphrodite{"26A5}
11552 \DeclareUTFSymbol\textNeutral{"26AA}
11553 \DeclareUTFSymbol\textPUuncrfemale{"26B2}
11554 \DeclareUTFSymbol\texthexstar{"26B9}
11555 \DeclareUTFSymbol\textSoccerBall{"26BD}
11556 \DeclareUTFSymbol\textSunCloud{"26C5}
11557 \DeclareUTFSymbol\textRain{"26C6}
11558 \DeclareUTFSymbol\textnoway{"26D4}
11559 \DeclareUTFSymbol\textMountain{"26F0}
11560 \DeclareUTFSymbol\textTent{"26FA}
11561 \DeclareUTFSymbol\textScissorRightBrokenBottom{"2701}
11562 \DeclareUTFSymbol\textScissorRight{"2702}
11563 \DeclareUTFSymbol\textScissorRightBrokenTop{"2703}
11564 \DeclareUTFSymbol\textScissorHollowRight{"2704}
11565 \DeclareUTFSymbol\textPhoneHandset{"2706}
11566 \DeclareUTFSymbol\textTape{"2707}
11567 \DeclareUTFSymbol\textPlane{"2708}
11568 \DeclareUTFSymbol\textEnvelope{"2709}
11569 \DeclareUTFSymbol\textPeace{"270C}
11570 \DeclareUTFSymbol\textWritingHand{"270D}
11571 \DeclareUTFSymbol\textPencilRightDown{"270E}
11572 \DeclareUTFSymbol\textPencilRight{"270F}
11573 \DeclareUTFSymbol\textPencilRightUp{"2710}
```

```

11574 \DeclareUTFSymbol\textNibRight{"2711}
11575 \DeclareUTFSymbol\textNibSolidRight{"2712}
11576 \DeclareUTFSymbol\textCheckmark{"2713}
11577 \DeclareUTFSymbol\textCheckmarkBold{"2714}
11578 \DeclareUTFSymbol\textXSolid{"2715}
11579 \DeclareUTFSymbol\textXSolidBold{"2716}
11580 \DeclareUTFSymbol\textXSolidBrush{"2717}
11581 \DeclareUTFSymbol\textPlusOutline{"2719}
11582 \DeclareUTFSymbol\textPlus{"271A}
11583 \DeclareUTFSymbol\textPlusThinCenterOpen{"271B}
11584 \DeclareUTFSymbol\textPlusCenterOpen{"271C}
11585 \DeclareUTFSymbol\textCross{"271D}
11586 \DeclareUTFSymbol\textCrossOpenShadow{"271E}
11587 \DeclareUTFSymbol\textCrossOutline{"271F}
11588 \DeclareUTFSymbol\textCrossMaltese{"2720}
11589 \DeclareUTFSymbol\textDavidStar{"2721}
11590 \DeclareUTFSymbol\textFourAsterisk{"2722}
11591 \DeclareUTFSymbol\textJackStar{"2723}
11592 \DeclareUTFSymbol\textJackStarBold{"2724}
11593 \DeclareUTFSymbol\textClowerTips{"2725}
11594 \DeclareUTFSymbol\textFourStar{"2726}
11595 \DeclareUTFSymbol\textFourStarOpen{"2727}
11596 \DeclareUTFSymbol\textFiveStarOpenCircled{"272A}
11597 \DeclareUTFSymbol\textFiveStarCenterOpen{"272B}
11598 \DeclareUTFSymbol\textFiveStarOpenDotted{"272C}
11599 \DeclareUTFSymbol\textFiveStarOutline{"272D}
11600 \DeclareUTFSymbol\textFiveStarOutlineHeavy{"272E}
11601 \DeclareUTFSymbol\textFiveStarConvex{"272F}
11602 \DeclareUTFSymbol\textFiveStarShadow{"2730}
11603 \DeclareUTFSymbol\textAsteriskBold{"2731}
11604 \DeclareUTFSymbol\textAsteriskCenterOpen{"2732}
11605 \DeclareUTFSymbol\textEightStarTaper{"2734}
11606 \DeclareUTFSymbol\textEightStarConvex{"2735}
11607 \DeclareUTFSymbol\textSixStar{"2736}
11608 \DeclareUTFSymbol\textEightStar{"2737}
11609 \DeclareUTFSymbol\textEightStarBold{"2738}
11610 \DeclareUTFSymbol\textTwelveStar{"2739}
11611 \DeclareUTFSymbol\textSixteenStarLight{"273A}
11612 \DeclareUTFSymbol\textSixFlowerPetalRemoved{"273B}
11613 \DeclareUTFSymbol\textSixFlowerOpenCenter{"273C}
11614 \DeclareUTFSymbol\textAsterisk{"273D}
11615 \DeclareUTFSymbol\textSixFlowerAlternate{"273E}
11616 \DeclareUTFSymbol\textFiveFlowerPetal{"273F}
11617 \DeclareUTFSymbol\textFiveFlowerOpen{"2740}
11618 \DeclareUTFSymbol\textEightFlowerPetal{"2741}
11619 \DeclareUTFSymbol\textSunshineOpenCircled{"2742}
11620 \DeclareUTFSymbol\textSixFlowerAltPetal{"2743}
11621 \DeclareUTFSymbol\textSnowflakeChevron{"2744}
11622 \DeclareUTFSymbol\textSnowflake{"2745}
11623 \DeclareUTFSymbol\textSnowflakeChevronBold{"2746}
11624 \DeclareUTFSymbol\textSparkle{"2747}
11625 \DeclareUTFSymbol\textSparkleBold{"2748}
11626 \DeclareUTFSymbol\textAsteriskRoundedEnds{"2749}
11627 \DeclareUTFSymbol\textEightFlowerPetalRemoved{"274A}
11628 \DeclareUTFSymbol\textEightAsterisk{"274B}
11629 \DeclareUTFSymbol\textCircleShadow{"274D}
11630 \DeclareUTFSymbol\textSquareShadowBottomRight{"274F}
11631 \DeclareUTFSymbol\textSquareTopRight{"2750}
11632 \DeclareUTFSymbol\textSquareCastShadowBottomRight{"2751}
11633 \DeclareUTFSymbol\textSquareCastShadowTopRight{"2752}
11634 \DeclareUTFSymbol\textDiamondSolid{"2756}
11635 \DeclareUTFSymbol\textRectangleThin{"2758}
11636 \DeclareUTFSymbol\textRectangle{"2759}
11637 \DeclareUTFSymbol\textRectangleBold{"275A}
11638 \DeclareUTFSymbol\textperp{"27C2}
11639 \DeclareUTFCommand\textnotperp{\textlstrikethru\textperp}
11640 \DeclareUTFSymbol\textveedot{"27C7}

```

```

11641 \DeclareUTFSymbol\textwedgedot{"27D1}
11642 \DeclareUTFSymbol\textleftspoon{"27DC}
11643 \DeclareUTFSymbol\textlbrackdbl{"27E6}
11644 \DeclareUTFSymbol\textlbrackdbl{"27E7}
11645 \DeclareUTFSymbol\textcirclearrowleft{"27F2}
11646 \DeclareUTFSymbol\textcirclearrowright{"27F3}
11647 \DeclareUTFSymbol\textlongleftarrow{"27F5}
11648 \DeclareUTFSymbol\textlongrightarrow{"27F6}
11649 \DeclareUTFSymbol\textlongleftarrow{"27F7}
11650 \DeclareUTFSymbol\textLongleftarrow{"27F8}
11651 \DeclareUTFSymbol\textLongrightarrow{"27F9}
11652 \DeclareUTFSymbol\textLongleftarrow{"27FA}
11653 \DeclareUTFSymbol\textlongmapsto{"27FC}
11654 \DeclareUTFSymbol\textLongmapsfrom{"27FD}
11655 \DeclareUTFSymbol\textLongmapsto{"27FE}
11656 \DeclareUTFSymbol\textnwsearrow{"2921}
11657 \DeclareUTFSymbol\textneswarrow{"2922}
11658 \DeclareUTFSymbol\textlhooknarrow{"2923}
11659 \DeclareUTFSymbol\extrhooknearrow{"2924}
11660 \DeclareUTFSymbol\textlhooksearrow{"2925}
11661 \DeclareUTFSymbol\extrhookswarrow{"2926}
11662 \DeclareUTFSymbol\textleadsto{"2933}
11663 \DeclareUTFSymbol\textrcurvearrowne{"2934}
11664 \DeclareUTFSymbol\textlcurvearrowse{"2935}
11665 \DeclareUTFSymbol\textlcurvearrowsw{"2936}
11666 \DeclareUTFSymbol\textrcurvearrowse{"2937}
11667 \DeclareUTFSymbol\textlcurvearrowdown{"2938}
11668 \DeclareUTFSymbol\textrcurvearrowdown{"2939}
11669 \DeclareUTFSymbol\textrcurvearrowleft{"293A}
11670 \DeclareUTFSymbol\textrcurvearrowright{"293B}
11671 \DeclareUTFSymbol\textleftrightharpoon{"294A}
11672 \DeclareUTFSymbol\extrightharpoon{"294B}
11673 \DeclareUTFSymbol\textupdownharpoonrightleft{"294C}
11674 \DeclareUTFSymbol\textupdownharpoonleftrightharpoon{"294D}
11675 \DeclareUTFSymbol\textleftleftharpoons{"2962}
11676 \DeclareUTFSymbol\textupupharpoons{"2963}
11677 \DeclareUTFSymbol\extrightrightarpoons{"2964}
11678 \DeclareUTFSymbol\textdowndownharpoons{"2965}
11679 \DeclareUTFSymbol\textleftbarharpoon{"296A}
11680 \DeclareUTFSymbol\textbarleftharpoon{"296B}
11681 \DeclareUTFSymbol\extrightbarharpoon{"296C}
11682 \DeclareUTFSymbol\textbarriharpoon{"296D}
11683 \DeclareUTFSymbol\textupdownharpoons{"296E}
11684 \DeclareUTFSymbol\textdownupharpoons{"296F}
11685 \DeclareUTFSymbol\textllparenthesis{"2987}
11686 \DeclareUTFSymbol\textrrparenthesis{"2988}
11687 \DeclareUTFSymbol\textinvdiameter{"29B0}
11688 \DeclareUTFSymbol\textobar{"29B6}
11689 \DeclareUTFSymbol\textobslash{"29B8}
11690 \DeclareUTFSymbol\textobot{"29BA}
11691 \DeclareUTFSymbol\textNoChemicalCleaning{"29BB}
11692 \DeclareUTFSymbol\textolessthan{"29C0}
11693 \DeclareUTFSymbol\textogreaterthan{"29C1}
11694 \DeclareUTFSymbol\textboxslash{"29C4}
11695 \DeclareUTFSymbol\textboxbslash{"29C5}
11696 \DeclareUTFSymbol\textboxast{"29C6}
11697 \DeclareUTFSymbol\textboxcircle{"29C7}
11698 \DeclareUTFSymbol\textboxbox{"29C8}
11699 \DeclareUTFSymbol\textValve{"29D3}
11700 \DeclareUTFSymbol\textmultimapboth{"29DF}
11701 \DeclareUTFSymbol\textshuffle{"29E2}
11702 \DeclareUTFSymbol\textupplus{"2A04}
11703 \DeclareUTFSymbol\textbigdoublewedge{"2A07}
11704 \DeclareUTFSymbol\textbigdoublevee{"2A08}
11705 \DeclareUTFSymbol\textJoin{"2A1D}
11706 \DeclareUTFSymbol\textfatsemi{"2A1F}
11707 \DeclareUTFSymbol\textcircplus{"2A22}

```

```

11708 \DeclareUTFSymbol\textminusdot{"2A2A}
11709 \DeclareUTFSymbol\textdottimes{"2A30}
11710 \DeclareUTFSymbol\textdttimes{"2A32}
11711 \DeclareUTFSymbol\textodiv{"2A38}
11712 \DeclareUTFSymbol\textinvneg{"2A3C}
11713 \DeclareUTFSymbol\textsqdoublecap{"2A4E}
11714 \DeclareUTFSymbol\textcapdot{"2A40}
11715 \DeclareUTFSymbol\textsqdoublecup{"2A4F}
11716 \DeclareUTFSymbol\textdoublewedge{"2A55}
11717 \DeclareUTFSymbol\textdoublevee{"2A56}
11718 \DeclareUTFSymbol\textdoublebarwedge{"2A5E}
11719 \DeclareUTFSymbol\textveedoublebar{"2A63}
11720 \DeclareUTFSymbol\texteqdot{"2A66}
11721 \DeclareUTFCommand\textneqdot{\textlstrikethru\texteqdot}
11722 \DeclareUTFSymbol\textcoloncolonequals{"2A74}
11723 \DeclareUTFSymbol\textleqslant{"2A7D}
11724 \DeclareUTFCommand\textnleqslant{\textlstrikethru\textleqslant}
11725 \DeclareUTFSymbol\textgeqslant{"2A7E}
11726 \DeclareUTFCommand\textngeqslant{\textlstrikethru\textgeqslant}
11727 \DeclareUTFSymbol\textlessapprox{"2A85}
11728 \DeclareUTFCommand\textnlessapprox{\textlstrikethru\textlessapprox}
11729 \DeclareUTFSymbol\textgtrapprox{"2A86}
11730 \DeclareUTFCommand\textngtrapprox{\textlstrikethru\textgtrapprox}
11731 \DeclareUTFSymbol\textlneq{"2A87}
11732 \DeclareUTFSymbol\textgneq{"2A88}
11733 \DeclareUTFSymbol\textlnapprox{"2A89}
11734 \DeclareUTFSymbol\textgnapprox{"2A8A}
11735 \DeclareUTFSymbol\textlesseqqgtr{"2A8B}
11736 \DeclareUTFSymbol\textgtreqqless{"2A8C}
11737 \DeclareUTFSymbol\texteqslantless{"2A95}
11738 \DeclareUTFSymbol\texteqslantgtr{"2A96}
11739 \DeclareUTFSymbol\textleftslice{"2AA6}
11740 \DeclareUTFSymbol\textright slice{"2AA7}
11741 \DeclareUTFSymbol\textpreceq{"2AAF}
11742 \DeclareUTFCommand\textnpreceq{\textlstrikethru\textpreceq}
11743 \DeclareUTFSymbol\textsucceq{"2AB0}
11744 \DeclareUTFCommand\textnsucceq{\textlstrikethru\textsucceq}
11745 \DeclareUTFSymbol\textprecneq{"2AB1}
11746 \DeclareUTFSymbol\textsuccneq{"2AB2}
11747 \DeclareUTFSymbol\textpreceqq{"2AB3}
11748 \DeclareUTFCommand\textnpreceqq{\textlstrikethru\textpreceqq}
11749 \DeclareUTFSymbol\textsucceqq{"2AB4}
11750 \DeclareUTFCommand\textnsucceqq{\textlstrikethru\textsucceqq}
11751 \DeclareUTFSymbol\textprecneqq{"2AB5}
11752 \DeclareUTFSymbol\textsuccneqq{"2AB6}
11753 \DeclareUTFSymbol\textprecapprox{"2AB7}
11754 \DeclareUTFCommand\textnprecapprox{\textlstrikethru\textprecapprox}
11755 \DeclareUTFSymbol\textsuccapprox{"2AB8}
11756 \DeclareUTFCommand\textnsuccapprox{\textlstrikethru\textsuccapprox}
11757 \DeclareUTFSymbol\textprecnapprox{"2AB9}
11758 \DeclareUTFSymbol\textsuccnapprox{"2ABA}
11759 \DeclareUTFSymbol\textsubseteqq{"2AC5}
11760 \DeclareUTFCommand\textnsubseteqq{\textlstrikethru\textsubseteqq}
11761 \DeclareUTFSymbol\textsupseteqq{"2AC6}
11762 \DeclareUTFCommand\textnsupseteqq{\textlstrikethru\textsupseteqq}
11763 \DeclareUTFSymbol\textdashV{"2AE3}
11764 \DeclareUTFCommand\textndashV{\textlstrikethru\textdashV}
11765 \DeclareUTFSymbol\textDashv{"2AE4}
11766 \DeclareUTFCommand\textnDashv{\textlstrikethru\textDashv}
11767 \DeclareUTFSymbol\textDashV{"2AE5}
11768 \DeclareUTFCommand\textnDashV{\textlstrikethru\textDashV}
11769 \DeclareUTFSymbol\textdownmodels{"2AEA}
11770 \DeclareUTFCommand\textndownmodels{\textlstrikethru\textdownmodels}
11771 \DeclareUTFSymbol\textupmodels{"2AEB}
11772 \DeclareUTFCommand\textnupmodels{\textlstrikethru\textupmodels}
11773 \DeclareUTFSymbol\textupspoon{"2AEF}
11774 \DeclareUTFSymbol\textinterleave{"2AF4}

```

```

11775 \DeclareUTFSymbol\textsslash{"2AFD}
11776 \DeclareUTFSymbol\textpentagon{"2B20}
11777 \DeclareUTFSymbol\textvarhexagon{"2B21}
11778 \DeclareUTFSymbol\textjinferior{"2C7C}
11779 \DeclareUTFSymbol\textslashdiv{"2E13}
11780 \DeclareUTFSymbol\textinterrobangdown{"2E18}
11781 \DeclareUTFSymbol\textfivedots{"2E2D}
11782 \DeclareUTFSymbol\textPUheng{"A727}
11783 \DeclareUTFSymbol\textPULhookfour{"A72C}
11784 \DeclareUTFSymbol\textPUscf{"A730}
11785 \DeclareUTFSymbol\textPUaolig{"A735}
11786 \DeclareUTFSymbol\textoo{"A74F}
11787 \DeclareUTFSymbol\textcircumlow{"A788}
11788 \DeclareUTFSymbol\textfi{"FB01}
11789 \DeclareUTFSymbol\textfl{"FB02}
11790 \DeclareUTFSymbol\textGaPa{"1D13B}
11791 \DeclareUTFSymbol\textHaPa{"1D13C}
11792 \DeclareUTFSymbol\textViPa{"1D13D}
11793 \DeclareUTFSymbol\textAcPa{"1D13E}
11794 \DeclareUTFSymbol\textSePa{"1D13F}
11795 \DeclareUTFSymbol\textZwPa{"1D140}
11796 \DeclareUTFSymbol\textfullnote{"1D15D}
11797 \DeclareUTFSymbol\texthalfnote{"1D15E}
11798 \DeclareUTFSymbol\textVier{"1D15F}
11799 \DeclareUTFSymbol\textAcht{"1D160}
11800 \DeclareUTFSymbol\textSech{"1D161}
11801 \DeclareUTFSymbol\textZwdr{"1D162}
11802 \DeclareUTFSymbol\textMundus{"1F30D}
11803 \DeclareUTFSymbol\textMoon{"1F319}
11804 \DeclareUTFSymbol\textManFace{"1F468}
11805 \DeclareUTFSymbol\textWomanFace{"1F469}
11806 \DeclareUTFSymbol\textFax{"1F4E0}
11807 \DeclareUTFSymbol\textFire{"1F525}
11808 \DeclareUTFSymbol\textBicycle{"1F6B2}
11809 \DeclareUTFSymbol\textGentsroom{"1F6B9}
11810 \DeclareUTFSymbol\textLadiesroom{"1F6BA}
11811 \DeclareUTFCommand\textcopyleft{\textcircled\textrevc}
11812 \DeclareUTFCommand\textccsa{\textcircled\textcirclearrowleft}
11813 \DeclareUTFSymbol\textglqq{"201E}
11814 \DeclareUTFSymbol\textgrqq{"201C}
11815 \DeclareUTFSymbol\textglq{"201A}
11816 \DeclareUTFSymbol\textgrq{"2018}
11817 \DeclareUTFSymbol\textflqq{"00AB}
11818 \DeclareUTFSymbol\textfrqq{"00BB}
11819 \DeclareUTFSymbol\textflq{"2039}
11820 \DeclareUTFSymbol\textfrq{"203A}
11821 \DeclareUTFSymbol\textneg{"00AC}
11822 \DeclareUTFSymbol\textcdot{"00B7}

11823 </xunextra>
11824 <@@=xeCJK>

```

5.22 xeCJK.cfg

```
11825 <*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。

```
11826
```

```
11827 </config>
```

版本历史

v3.1.0 (2012/11/13 – 2012/11/21)

- General: 放弃对 `\outer` 宏的特殊处理。..... 1
- 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。 142
- 改用 `indentfirst` 宏包处理缩进的问题。..... 148
- 取消 `\cprotect` 的外部宏限制。..... 166
- 删除多余的 `default-itcorr` 结点。..... 49
- 使用 `xtemplate` 宏包的机制来组织标点符号的处理。.. 111
- LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。..... 147
- `\xeCJK@fix@penalty`: 采用通过不修改原语 `\` 的方式对修复倾斜校正。..... 153
- `\_xeCJK_fallback_loop:n`: 调整备用字体的循环方式。..... 123
- `\xeCJK_glyph_if_exist:N`: 改进 `fontspec` 宏包中定义的 `\font_glyph_if_exist:NnTF`。..... 25
- `\xeCJK_hook_for_ulem::`: 简化对 `ulem` 宏包的兼容补丁。 167
- `\c_xeCJK_space_skip_tl`: 字间空格考虑 `\spaceskip` 不为零的情况。..... 25
- `\_xeCJK_switch_font:n`: 改进定义, 加快切换速度。 . 131
- `\xeCJKVerbAddon`: 新增 `\xeCJKVerbAddon` 用于抄录环境中的间距调整。..... 143

v3.1.1 (2012/12/02 – 2012/12/13)

- General: 不再依赖 `xpatch` 宏包。..... 1
- 对于与 `xltextra` 的冲突给出错误警告。..... 152
- 增加 `NewLineCS` 和 `EnvCS` 选项。..... 105
- 增加小宏包 `xeCJKintef`, 用于处理下划线的问题。.... 167
- CheckFullRight: 处理全角右标点之后的断行问题。... 100
- InlineEnv: 改变行内环境的设置方式, 从而使用 `\str_case:x:nnn` 代替原来的 `\clist_if_in:NnTF` 来判断是否是行内环境。..... 105
- PlainEquation: 增加 PlainEquation 选项。..... 104
- `\xeCJK@family`: 修改主要 CJK 字体族的自动更新方式。 . 152
- `\_xeCJK_check_single_aux:n`: 改进定义, 减少使用 `peek` 函数的次数。..... 103
- `\_xeCJK_check_single_space:NN`: CheckSingle 支持段末“汉字 + 汉字 + 空格 + 汉字/标点”的形式。.... 104
- `\xeCJK_hook_for_ulem::`: 完全处理下划线里的标点符号的有关问题。..... 167
- `\xeCJK_peek_catcode_ignore_spaces:NTF`: 新增有省略空格标识的 `peek` 函数。..... 27
- `\xeCJK_save_class:n`: 使用 `\xeCJK_save_class:n` 保存 `XgTeX` 预定义的字符类别。..... 29
- `\xeCJK_set_char_class:nnn`: 在文档中设置字符类别时不重复设置 `\catcode`。..... 37
- `\_xeCJK_set_char_class_eq:n`: 交换参数的顺序。... 37
- `\_xeCJK_set_verb_exspace::`: 调整间距的计算方法。 . 144
- `\xeCJKnobreak`: 增加 `\nobreak` 的 `xeCJK` 版本。..... 102

v3.1.2 (2012/12/27 – 2013/01/01)

- General: 解决在下划线状态下使用 `\makebox` 时的错误。 175
- 修正非 `\UTFencname` 编码下面 `xunicode` 重定义的 `\nobreakspace` 会失效的问题。..... 152
- 修正重定义 `\CJKfamilydefault` 无效的问题, 恢复容错能力。..... 138

\xeCJK@family: 不将参数完全展开。..... 152

- `\_xeCJK_check_single_space:NN`: 使用 `\xeCJK_if_CJK_class:NTF` 来代替 `\int_case:nnn` 判断是否是 CJK 字符类。..... 104
- `\_xeCJK_family_unknown_warning:n`: 在没有定义任何 CJK 字体的情况下, 不再重复给出字体没有定义的警告。 134

v3.2.0 (2013/04/14 – 2013/05/22)

- General: 增加 `IVS` 字符类用于处理异体字选择符。..... 29
- 增加 `Verb` 选项。..... 142
- `\setCJKmonofont`: 定义中加入 `\normalfont`。..... 135
- `\_xeCJK_Boundary_and_FullLeft_glue:N`: 当全角左标点前面是 `hlist`、`none`、`glue` 和 `penalty` 等节点时, 压缩其左空白。..... 94
- `\l_xeCJK_family_tl`: 不将其初始化为 `\CJKfamilydefault`。..... 134
- `\xeCJK_FullLeft_and_Default::`: 修正 `xeCJK` 使西文在部分情况下无法断词的问题。..... 93
- `\c_xeCJK_space_skip_tl`: 字间空格考虑到 `\spacefactor` 和 `\xspaceskip` 的情况。..... 25

v3.2.1 (2013/05/29)

- General: 调整 `Verb` 选项: 在命令 `\verb` 里使用时, 不破坏标点禁则, 增加值 `env+`。..... 142

v3.2.2 (2013/05/30 – 2013/06/04)

- General: 修正某些重音不能正确显示的问题。..... 1
- 增加小宏包 `xeCJK-listings`, 用于支持 `listings` 宏包。.. 193
- `\_xeCJK_ulem_CJK_and_FullRight_glue:N`: 修正下划线不能跳过全角右标点的问题。..... 178

v3.2.3 (2013/06/04 – 2013/06/11)

- General: 不再改变 CJK 字符类的 `\catcode`。..... 37
- 根据 `XgTeX` 的脚本重新整理全角标点符号。..... 31
- 解决 CheckSingle 选项与 `tablists` 宏包的冲突。..... 104
- 提供四个 `TECkit` 映射文件用于句号转换和简繁互换。... 1
- 完善对 `listings` 宏包的支持。..... 193
- `\_xeCJK_listings_initial_hook::`: 解决 `listings` 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。..... 194
- `\_xeCJK_listings_process_Default:n`: 在 `listings` 环境中对 `\charcode` 大于 255 的字符根据其 `\catcode` 区分 `letter` 和 `other`。..... 195
- `\_xeCJK_restore_shipout_CJKsymbol::`: 解决 `\CJKunderdot` 跨页使用时影响到页眉页脚的问题。... 192
- `\_xeCJK_ulem_FullLeft_and_CJK::`: 修正全角左标点后下划线与 `\CJKunderdot` 连用时结果不正常的问题。.. 179
- `\xeCJKVerbAddon`: 新增 `\xeCJKOffVerbAddon` 用于局部取消 `\xeCJKOffVerbAddon` 的影响; 并解决跨页使用时影响到页眉页脚的问题。..... 143

v3.2.4 (2013/06/23 – 2013/07/06)

- General: 不再使用 `CJKnumber` 选项, 可以在 `xeCJK` 之后直接使用 `CJKnumb` 宏包得到中文数字。..... 166
- 改进获取分区字体属性的办法。..... 126
- 解决使用 CheckSingle 时, 某些 `\CJKglue` 不能被正确加入的问题。..... 104
- 尽量移除用作判断标志的 `\kern`。..... 49
- 内部调整分区字体的设置方法。..... 126

使 listings 的 breaklines 选项对 CJK 字符类可用, 并保持标点符号的禁则。	196	修正 unicode-letters.tex 中谚文符号 \catcode 不准的问题。	37
使用 AllowBreakBetweenPuncts 时, 相应标点符号仍能与边界对齐。	93	\Url@MathSetup: 使通过 \UrlFont 等命令设置的 CJK 字体生效。	150
修正 xeCJKfntef 与 natbib 等的冲突。	167	_xeCJK_check_single_aux:n: 与 \CJKspace 兼容。	103
遵循 L ^A T _E X3 变量需要预先声明的原则。	1	_xeCJK_punct_glue:NN: 标点符号左/右空白的伸展值不超过原始边界, 收缩值不小于另一侧边界。	92
\addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。	136	\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。	139
CJKfilltwosides: 改用 minipage 和 L ^A T _E X 表格 (tabular) 来实现。	193	v3.2.8 (2013/11/16 – 2013/12/05)	
_xeCJK_Boundary_and_FullLeft_glue:N: 细化边界与全角左标点之间是否压缩空白的判断。	94	General: 启用 xunicode 中的带圈数字和字母设置。	212
_xeCJK_fallback_loop:n: 使 \CJKfamilydefault 的 FallBack 设置全局可用。	123	\DeclareUTFmathsymbols: 修正 \UseMathAsText 的功能, 恢复 \hbar 和增加以 text 打头的文本符号命令。	201
_xeCJK_set_verb_exspace:: 当计算得出的间距为负时, 缩小 CJK 字体。	144	_xeCJK_nobreak_skip:: 禁止在 \verb 中断行。	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	\xeCJKVerbAddon: 增加是否是等宽字体的判断。	143
\xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。	27	v3.2.9 (2013/12/07 – 2013/12/08)	
v3.2.5 (2013/07/10 – 2013/07/25)		General: 文档部分增加 xunicode 定义的符号表。	200
General: 恢复 \nobreakspace 的原始定义。	152	增加 xunicode-extra.def 中, 用于加入 puenc.def 中的符号定义。	215
解决 fixltx2e 和 amsthm 的冲突。	152	\DeclareEncodedCompositeAccents: 修正 xunicode 中的错误定义。	206
修正 CJK 和 NormalSpace 字符类之间因为边界造成的间距不正确的问题。	45	\c__xeCJK_middle_dot_prop: 完整处理 encguide.pdf 的编码符号表中, 与旧编码的 U+00B7 冲突。	156
增加小宏包 xunicode-addon, 为 xunicode 提供判断字符是否存在的功能。	200	v3.2.10 (2014/02/20 – 2014/03/01)	
\@setupverbvisiblespace: 可视空格考虑传统 T _E X 字体的情况。	145	\CJKaddEncHook: 使用 CJKnumb 时, 让 \Unicode 有定义。	166
Verb: 微调定义。	142	\DeclareUTFDoubleEncodedAccent: 改进 \t 等的定义方式。	206
_xeCJK_Boundary_and_FullLeft_glue:N: 细化全角左标点是否位于段首的判断。	94	\DeclareUTFDoubleEncodedSymbol: 改进 \sliding 等的定义方式。	206
增加对 enumitem 宏包修改的 \item 的判断。	94	\DeclareUTFTIPACCommand: 检查 \t 和 \sliding 的参数是否以 \textipa 开头。	210
_xeCJK_boundary_register_text_command:NN: 解决汉字后紧跟 \(\dots\) 形式的行内数学公式时, 不能加入间距的问题。	150	LoadFandol: 当没有设置字体时, 使用 Fandol 字体系列。	137
\xeCJKVerbAddon: 禁止自动换行, 与西文一致。	143	v3.2.11 (2014/03/14 – 2014/04/10)	
v3.2.6 (2013/07/29 – 2013/08/15)		General: 删除 \xeCJKcaption。	166
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。	127	左右角括号 U+2329 和 U+232A 是西文标点符号。	31
case 类函数的用法与 L ^A T _E X3 同步。	1	\CJK@family: 引入 \CJK@family 保存实际的字体族名。	134
为 \mathrm 减少一个可能的数学字体族。	150	indentfirst: 放弃 indentfirst 和 CJKnumber 选项。	147
\AtEndUTFCommand: 可以指定特定符号命令使用的钩子。	209	\xeCJK_add_to_shipout:n: 不再使用内部名字。	24
_xeCJK_boundary_register_text_command:NN: 考虑 ulem 对 \MakeRobust 的不当定义。	150	v3.2.12 (2014/05/12)	
考虑 \math 和 \ensuremath。	150	General: 更新 \int_to_Hex:n。	123
\xeCJK_CJK_and_Boundary:w: 更好的处理边界是 \relax 的情况。	85	新增 RubberPunctSkip 选项。	108
\xeCJK_set_mathfont:: 设置粗体时先检查对应字体是否存在。	139	v3.2.13 (2014/06/02 – 2014/06/20)	
v3.2.7 (2013/08/22 – 2013/11/09)		General: 自动调整 \CJKfamilydefault 时, 只将 \familydefault 展开一次。	138
General: 标点宽度设置禁用比例选项的值改为 nan。	118	\xeCJK_set_mathfont:: 修复参数类型错误。	139
处理 AllowBreakBetweenPuncts 与 xeCJKfntef 的兼容问题。	93	v3.2.14 (2014/10/31 – 2014/11/03)	
实现自定义行首/尾标点符号宽度功能。	107	General: xeCJKfntef 不再依赖 CJKfntef。	167
使用 everypage 往 \shipout 盒子里加钩子。	24	解决下划线前后没有 \CJKglue 或 \JKEcglue 的问题。	167
		完善 \varCJKunderline 的实现。	167
		v3.2.15 (2014/11/07 – 2014/11/10)	
		General: xeCJKfntef 增加 hidden 选项。	167
		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。	187

<code>_xeCJK_listings_initial_hook::</code> 修正 breaklines 无效的问题。..... 194 <code>\xeCJKfontofon</code>: 完善选项。..... 185 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>。..... 150 整理 <code>xCJKkecglue</code> 的部分代码。..... 84 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。..... 152 新选项 <code>WidowPenalty</code>。..... 102 <code>\CJKaddEncHook</code>: 应用 0.99992 版的新原语 <code>\Ucharcat</code>。..... 166 <code>LoadFandol</code>: 为方便 MacTeX 用户, <code>Fandol</code> 字体改用文件名。..... 137 <code>_xeCJK_boundary_register_text_command:NN</code>: 兼容 $\text{\LaTeX} 2_{\epsilon}$ 2015。..... 150 <code>\xeCJK_check_single_cs:NNn</code>: 补充可能遗漏的空格。..... 104 <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。..... 194 <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。..... 195 <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 $\text{\LaTeX} 2_{\epsilon}$ 2015。..... 154 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。..... 154 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 $\text{\LaTeX} 3$ 代码。..... 1 兼容 $\text{\LaTeX} 2_{\epsilon}$ 2016/02/01 的字符类设置。..... 30 解决与 <code>microtype</code> 宏包的兼容问题。..... 166 使用新的 Unicode 编码名称 TU。..... 200 <code>CJKfilltwosides</code>: 确保进入水平模式。..... 193 v3.3.4 (2016/02/07) General: 兼容 $\text{\LaTeX} 0.99994$ 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>。..... 108 <code>CJKmath</code> 功能也支持分区字体。..... 140 标点符号的压缩量能伸长到原始空白, 能收缩到较小边距。..... 119 改进 <code>xCJKkecglue</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。..... 139	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>。..... 167 修复 <code>CJKspace</code> 功能失效。..... 85 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。..... 135 <code>\setCJKfallbackfamilyfont</code>: 允许字体属性可选项在后的新语法。..... 125 <code>\setCJKmathfont</code>: 允许字体属性可选项在后的新语法。..... 139 <code>\setCJKmonofont</code>: 允许字体属性可选项在后的新语法。..... 135 <code>_xeCJK_long_punct_kerning:N</code>: 考虑破折号边界为负值的情况。..... 113 v3.4.4 (2016/11/30) General: 不压缩长标点与其他标点的间距。..... 118 v3.4.5 (2017/01/02) General: 更新 $\text{\LaTeX} 3$ 的过时用法。..... 150 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。..... 152 v3.4.7 (2017/03/20) General: 简化 <code>CheckSingle</code> 的实现, 不再展开宏。..... 103 v3.4.8 (2017/05/15) General: 转义 <code>\lstinline</code> 参数中的 $\text{\LaTeX}$。..... 200 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 ($\text{\LaTeX} 3$ 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。..... 123 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>。..... 99 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。..... 97 同步 $\text{\LaTeX} 3$ 2017/12/16。..... 1 新增 <code>PunctFamily</code> 选项支持对汉字标点单独切换字体。..... 133 修正标点同为 <code>LongPunct</code> 与 <code>MiddlePunct</code> 时的实现错误。..... 99 总允许长标点与其他标点之间折行。..... 92 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。..... 97 v3.7.0 (2018/03/12 – 2018/03/18) General: 补充定义 <code>\textthyphenationpoint</code> 和 <code>\texttwoemdash</code>。..... 212 不再默认引入 <code>xunicode</code> 宏包。..... 148 对 <code>\nobreakspace</code> 的恢复放到 <code>xunicode-addon</code> 中处理。..... 152
---	---

修正长标点被隔开时的压缩处理错误。.....	113	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。...	209	\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。.....	192	同步 Unicode 14.0。.....	33
简化 CJKspace 的实现, 并修复错误。.....	85	v3.8.9 (2022/05/26)	
解决与 microtype 宏包的兼容问题。.....	212	General: 修正居中标点悬挂错误。.....	117
删除定义新字体族时过滤重复选项的功能。.....	128	_xeCJK_bound_type_1_glue:Nn: 增加位于段首的支架	
同步 L ^A T _E X3 2019/03/05。.....	1	盒子判断。.....	95
\xeCJK_FullLeft_and_Default:: 再次修正 FullLeft 类		v3.9.0 (2022/06/06 – 2022/07/08)	
字符与西文连用断词失败的问题。.....	93	General: 不直接依赖 xparse 和 l3keys2e。.....	23
_xeCJK_patch_tuenc_composite:: 修复补丁错误。..	156	修复西文的 character protrusion 功能。.....	84
v3.7.3 (2019/04/15)		v3.9.1 (2022/07/28 – 2022/08/02)	
General: 补充日文假名扩展。.....	33	General: 简化部分内部实现。.....	49
修复 penalty 数值错误。.....	97	修复下划线中数学公式的错误处理。.....	176
v3.7.4 (2019/05/31)		v3.10.0 (2022/08/31 – 2026/06/23)	
General: 简化行首/尾标点符号宽度的实现。.....	107	General: NoBreakLongPunct 在右侧时禁止折行。.....	92
v3.8.0 (2020/02/09 – 2020/02/10)		\reset@color 补丁增加 hlist 回退路径, 当 color pop 后	
General: 兼容 L ^A T _E X 2 _ε 2020/02/02 对 NFSS 的修改。.....	1	最后节点为 hlist 时设置	
清理过时的兼容性补丁代码。.....	1	\g__xeCJK_reset_color_pending_bool, 延迟到	
删除 CJKfntef 补丁。.....	166	_xeCJK_check_for_glue_skip: 中处理 \colorbox	
删除 realscripts 补丁。.....	152	等命令右侧间距(#831)。.....	161
删除 \hbar 补丁。.....	150	补丁 l3color 后端的 _color_select:N 和	
删除 \mathrm 补丁。.....	150	_color_backend_reset:, 使 l3color 接口的颜色切换	
应用 \fp_if_nan:nTF。.....	118	也能正确保持 xeCJK 间距(#832)。.....	162
应用 \peek_remove_spaces:n。.....	27	补丁 \reset@color 以在颜色弹出 whatsit 之后重新放	
\@setupverbvisiblespace: 更新可视空格补丁。.....	145	置 xeCJK 节点标记, 修复 \textcolor 命令右侧多余	
v3.8.1 (2020/02/14)		inter-word glue 的问题(#831)。.....	161
General: 修复 \l_xeCJK_current_font_tl 标记错误。..	130	撤回对 \char 的重定义, 改为提供 \xeCJKchar 新命令,	
应用 \shapedefault。.....	140	避免破坏依赖	
v3.8.2 (2020/02/17)		let	
General: 避免导言区字体警告。.....	135	保存 \tn {char} 原语的宏包(\#800)。hdclindex8480157	
修复分区字体错误。.....	131	改正拼写错误 _xeCJK_glue_node:n。.....	49
v3.8.3 (2020/03/15 – 2020/04/27)		将全角浪线等连接号从 MiddlePunct 改为 LongPunct,	
General: hidden 选项保留原内容的高度和深度。.....	174	消除不必要的标点压缩间距。.....	31
补充 U+02EA 和 U+02EB。.....	33	使用 \l_keys_key_str 和 \l_keys_choice_str 替代	
兼容 unicode-math 和 CJKmath 选项。.....	158	已废弃的 _tl 版本(#806)。.....	161
取消 xeCJKfntef 的初始彩色设置。.....	191	提升 L ^A T _E X3 最低版本要求至 2025/10/09。.....	22
删除 _nopar。.....	1	提升版本号至 v3.10.0。.....	1
同步 Unicode 13.0。.....	33	通过在 \Hy@BeginAnnot 中保存并选择性恢复 xeCJK 节	
修复 xCJKecglue 选项。.....	97	点标记, 同时清除旧标记, 解决目录中链接注释起始处的	
依赖 ctexhook 宏包。.....	24	虚假 ecglue(#810), 并为 \ref 提供前侧 ecglue(#809)。159	
重构 PunctStyle 选项, 完全展开参数。.....	121	同步 Unicode 15.0。.....	33
v3.8.4 (2020/05/31 – 2020/06/04)		同步 Unicode 15.1。.....	33
General: 重构后备字体的实现, 修正标点符号无后备字		同步 Unicode 17.0。.....	33
体的问题。.....	122	为 color/xcolor 添加兼容补丁, 在颜色切换 whatsit 后重	
\xeCJK_font_gset_to_current:N: 不缓存 \nullfont。..	25	放 xeCJK 节点标记, 修复 \textcolor 后 CJKecglue 丢	
v3.8.5 (2020/06/25 – 2020/06/26)		失的问题(#315, #803)。.....	161
General: 进一步兼容 microtype。.....	159	为 hypdoc 的 \HD@target 添加补丁, 在它产生的 hbox	
_xeCJK_bound_type_1_glue:Nn: 增加盒子高度判断。..	95	之后重放 xeCJK 节点标记, 修复 l3doc 中 \cs_meta 等	
v3.8.6 (2020/10/17 – 2020/10/18)		命令后 CJKecglue 丢失或保留为原始空格的问题	
General: 兼容 L ^A T _E X 2020/10/01 的 NFSS 钩子机制。...	152	(#873)。.....	162
正确还原标点符号后的 penalty 状态。.....	97	为 mtp ₂ 提供兼容补丁, 使大花括号内部的 \char 不	
v3.8.7 (2021/06/04 – 2021/06/09)		被 interchar 拦截(#407)。.....	158
General: 更好地兼容 CJKnumb。.....	166		

为 url 的 \Url@FormatString 添加补丁, 在进入数学模式前 drain 缓存的 CJKecglue, 修复 CJK 文字与 \url 命令之间间距丢失的问题(#880)。	163	修复 CJK 字符后行尾空格在宏命令前未被消除的问题, 不再立即输出空格 glue, 改为只留标记由后续恢复路径决定间距(#324)。	85
文档字体从 Source Han Serif 统一为 Noto Serif CJK (#686)。	1	\xeCJK_fntef_sbox:n: 保存并恢复	
新增 NoBreakLongPunct 属性, 禁止在省略号等长标点前断行。	99	\g__xeCJK_last_node_tl, 避免 hbox 内渲染装饰符号时 interchar toks 污染全局节点标记, 修复 xeCJKfntef 与 \textcolor 组合使用时多余的 CJKecglue。	189
新增 \xeCJKchar, 绕过 interchar 机制(#407)。	157	\xeCJK_make_node:n: 新增	
新增实验性 experiment/halfright-prebreakpenalty 选项(#811)。	88	\g__xeCJK_glue_check_pending_bool 用于标记 xeCJKfntef 组结束后的首次 glue 探测。	50
修复 xCJKecglue 错误。	49	新增 \g__xeCJK_reset_color_pending_bool 用于标记 \reset@color 在 hlist 上下文中触发, 延迟到	
修复 \set@color 补丁在无真实节点标记时未清除全局状态, 导致首次 \textcolor 在标点或段首后插入虚假 ecglue 的问题(#807)。	161	__xeCJK_check_for_glue_skip: 中处理 \colorbox 等命令右侧的 color pop whatsit(#831)。	50
修复 \textcolor 包裹 ulem 类下划线命令时 CJK 字间距异常的问题(#830)。	1	\c__xeCJK_middle_dot_prop: 在 \Pifont 中先进入水平模式, 防止 \makeCJKinactive 在垂直模式下通过分页泄漏到输出例程(#688)。	157
修复 \textcolor 等 whatsit 节点导致 CJKecglue 丢失的问题。	49	__xeCJK_punct_boundary_guard:: 新增	
修复 \textnleqslant 的定义, 修复 \textnbacksim 和 \textnlessapprox 定义里的自引。	215	experiment/punct-measure-fix 选项: 段落模式下通过 para/end 钩子恢复被 \unskip 移除的标点补偿 glue (#859)。	89
修复 \textsbleftarrow 的定义。	212	修复 tabular 中 \unskip 吞掉标点补偿 glue 的问题 (#827)。	89
修复 \xeCJKnobreak 错误。	49	__xeCJK_skip_if_interword:N: 移除通用 whatsit 恢复, 避免 hyperref 链接注释等 whatsit 节点导致 CJKecglue 在错误位置插入(#803)。配合 \set@color 定点补丁保持 \textcolor 场景的正确性(#315)。	42
修复拼错的命令名, \cyrreref → \cyrrerev 和 \textDiamandSolid → \textDiamondSolid。	215	\xeCJK_under_symbol:nnnnn: 在独立模式的末尾设置 \g__xeCJK_glue_check_pending_bool, 修复 \CJKunderdot 等命令右侧空格问题(#826)。	191
修复颜色补丁在 \g__xeCJK_last_node_tl 被无关的 \set@color 清空后仍尝试重放节点导致 Missing number 错误的问题, 影响 listings 与 \rulecolor 组合 (#836)。	161	\xeCJKsetcharclass: Deprecated:	
修复在 \lstinline 参数中把非 # 的字符(如 &)用 \catcode 设为 catcode 6 时输出错字的问题(#879)。	193	\xeCJKsetcharclass 已弃用, 调用时报错并提示改用 \xeCJKDeclareCharClass(#709)。	149
修复字体切换组(如 \texttt、\zihao)导致 CJKecglue 使用错误字体度量的问题。	49	v3.10.1	(2026/06/29 – 2026/07/01)
修复组合符号 U+04AA 的定义。	215	General: 提升版本号至 v3.10.1。	1
在下划线内 CJK 分组切换时保存/恢复字体状态 (#465)。	176	为 l3doc 的 __codedoc_meta:n 添加补丁, 将参数内容包入 \hbox:n, 修复 \meta 内首字符前出现多余 CJKecglue 的问题(#920)。	164
\CJKfontspec: Breaking: \newCJKfontfamily 定义的字体切换命令改为局部定义, 与 \newcommand 行为一致。在分组内调用时, 命令不再泄漏到分组外(#751)。	135	为 L ^A T _E X 核心命令 \verb 添加补丁, 在进入 verb 内容前 drain 缓存的 CJKecglue, 修复 CJK 文字与 \verb(或 shortverb 的短记号)之间间距丢失的问题(#910)。	163
\CJKunderanysymbol: 移除 \ignorespaces, 修复命令后空格丢失(#465)。	191	为 bibl _{at} ex 的 \blx@pagetracker 添加补丁, 清空 \g__xeCJK_last_node_tl, 修复中文参考文献条目首字符前出现多余 CJKecglue 的问题(#931)。	166
\CJKunderdot: 移除 \ignorespaces, 修复命令后空格丢失(#465)。	191	v3.10.2	(2026/07/03 – 2026/07/06)
\xeCJK@family: Breaking: 移除对 L ^A T _E X 2 _ε 2020/10/01 之前版本的字体钩子兼容代码(#746)。	152	General: PoZheHao 类字符同样按全角右标点处理(#382)。	110
\xeCJK_check_for_glue:: 当 glue 下方是 hlist(\mbox 等命令产生的 hbox)时, 不依赖 boolean, 直接通过 \g__xeCJK_last_node_tl 判断前方内容类型, 修复 \mbox、\colorbox 等 hbox 命令右侧多余 inter-word glue(#831)。	46	\verb 组关闭时用 \setlanguage 主动排出 language whatsit, 修复 \verb 右侧源码空格场景下 CJKecglue 双倍的问题(#919)。	164
新增 glue 分支处理, 修复 xeCJKfntef 命令右侧空格问题。	46	提升版本号至 v3.10.2。	1
\xeCJK_CJK_and_Boundary:w: 当显式} 触发 CJK→Boundary 转换时, 设置 \g__xeCJK_glue_check_pending_bool 以启用后续的 glue 检查, 修复前 {中} 后类型排版中多余的 \CJKecglue(#831)。	85	为 doc 的 \meta 添加补丁, 将参数内容包入 \hbox:n, 修复 \meta 内首字符前出现多余 CJKecglue 的问题 (#951)。	164
		文档等宽字体改用朱雀仿宋, 意大利体改用霞鹜文楷 GB Lite(#908)。	1

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)。	163
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)。	108	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)。	159
新增 PoZheHaoLigature 选项, 支持字体的破折号合字功能(#382)。	108	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)。	162
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)。	92	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)。	161
_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)。	113	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)。	115	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)。	115	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)。	163
\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)。	163
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)。	122	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, 并保留左标点后接右标点的自然空白(#975)。	122	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	包装内核私有 \@imakebox 与 \@ifmakebox 的 [1] [2] #3 签名以注册 \makebox / \framebox; 内核若调整该签名须同步此 wrapper(#992)。	83
新增 CJLineBreak 选项, 为日文小假名提供可选严格行首禁则(#165)。	108	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 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 enabled-left-right-kerning 标点格式选项(#975)。	14	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
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)。	159	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
\CJKsout: 使用 \cleaders 使删除线相对正文居中(#967)。	187	\xeCJK_check_for_glue:: Boundary 到 CJK 的源码空格检查改用与 Default 方向相同的	
\CJKKunderanyline: 使用 \cleaders 使自定义线相对正文居中(#967)。	187	_xeCJK_skip_if_interword:N 判断函数, 自然宽度	
\CJKKunderdblline: 使用 \cleaders 使双下划线相对正文居中(#967)。	186		
\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)。	185		
\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)。	186		
\CJKxout: 使用 \cleaders 使斜删除线相对正文居中(#967)。	187		
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)。	166		
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)。	162		

- 不等于当前词间空格的显式 glue 不再被替换为 `\CJKglue(#996)`。..... 46
- 由命令边界 framework 接管 `\xeCJKfntef`、盒子命令和颜色命令, 删除按全局 `\tl` 猜测 `\hlist/whatsit` 下方类别的旧回退; glue 分支现在只接受紧邻的 `\xeCJK marker(#992)`。46
- `\xeCJK_fntef_sbox:n`: 改用 framework 的可嵌套暂停区隔离测量装饰符号时构造的盒子, 同时保存、恢复 marker 与 pending 状态(#992)。..... 189
- `\_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)。..... 199
- `\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)。..... 183
- 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)。..... 183
- 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)。.. 164
- 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)。.. 159
- `\_xeCJK_boundary_replay_before::` `\sbox` 改用专用适配器: `\global` 等前缀必须紧邻 `\setbox`, 放进 `cmd/sbox/before` 钩子的赋值会把前缀吃掉, 使 `\global\sbox` 静默退化为局部赋值(#1029)。..... 83
- boundary 判断前把参数副本中 catcode 4 的 & 换成 `\scan_stop:`, 修复 `\colorbox` 等命令在 `\halign` 语境下参数含 .. & 时编译报错的问题(#1043)。hdclindex192553
- 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)。... 82
- `\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)。..... 86
- `\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)。..... 185
- `\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)。..... 189
- `\_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)。..... 183
- 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)。..... 183
- `\_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)。..... 173
- `\_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)。..... 181
- v3.10.6** (2026/08/11 – 2026/08/20)
- General: 花括号分组内的间距改用「盒内 kern 占位加词尾补伸缩」的方式输出, 使 `\CJKecglue` 的收缩量与 `\CJKglue` 的伸长量都回到外层列表(#1067)。..... 181
- `\_xeCJK_boundary_replay_before::` post-transparent 搬移「marker + 候选 glue」后缀前, 先用 `\skip_if_finite:nTF` 排除 `\hfill/\hfil` 一类无限阶填充 glue, 使它们不再被误搬到零尺寸盒子之后, 修复 CJK 文字紧接源码空格与 `\hfill\hfil` 时两侧填充不对称(#1085)。..... 62

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

\&	1426
\(	1459
\)	1491
\\	5, 6, 7, 13, 14, 15, 16, 31, 366, 367
\(font-switch)	9

A

\addCJKfontfeatures	10
\AddToHook	2834, 2836, 2847, 2849, 2860, 2862, 2873, 2883, 2885
AllowBreakBetweenPuncts	6
\AtBeginDocument	70
\AtEndOfPackage	431, 727, 740, 751, 775, 805
AutoFakeBold	4, 9
AutoFakeSlant	4, 9
AutoFallback	4

B

bool commands:

\bool_gset_false:N	929, 945, 957, 1093, 1103, 1132, 1149, 1733, 1854, 2132, 2232, 2365, 2468, 2586
\bool_gset_true:N	1732, 2827, 3085, 3368
\bool_if:NTF	102, 320, 608, 610, 631, 902, 944, 1080, 1090, 1094, 1101, 1109, 1119, 1131, 1135, 1148, 1180, 1713, 2049, 2055, 2332, 2930, 2949, 2951, 3033, 3043, 3344, 3388
\bool_lazy_and:nnTF	2382
\bool_lazy_or:nnTF	555
\bool_lazy_or_p:nn	2388
\bool_new:N	53, 111, 270, 613, 614, 635, 725, 1309, 1317, 1318, 1319, 3316, 3336
\bool_set_false:N	245, 2933, 2934, 3290, 3328
\bool_set_true:N	104, 252, 721, 2942, 2947, 2965, 2971, 2989, 2995, 3005, 3018, 3027, 3037, 3047, 3279, 3301, 3322

Boundary	369
----------	-----

box commands:

\box_clear:N	2412, 2444
\box_dp:N	1914, 2395
\box_ht:N	1913, 2391
\box_if_exist:N	1336
\box_new:N	51, 1325, 1326, 1327, 1328, 1337, 2372
\box_set_to_last:N	1909, 2356
\box_use_drop:N	1918, 1938, 1941, 1961, 1967, 1971, 1978, 2499, 2522
\box_wd:N	224, 1912, 2385
boxdepth	18

C

cctab commands:

\cctab_select:N	54
\c_code_cctab	54

char commands:

\char_generate:nn	434
\char_set_catcode_alignment:N	54, 1426
\char_set_catcode_ignore:n	332, 333, 334, 335, 336
\char_set_catcode_letter:n	573
CheckFullRight	6
CheckSingle	4
CJK	369
\CJKecglue	839, 1004, 1398, 1402, 2024, 3390
CJKecglue	3, 3269
\CJKfamily	9
\CJKfamilydefault	10
CJKfilltwosides	19
\CJKfontspec	10
\CJKglue	803, 1111, 1121, 1157, 1161, 1183, 1226, 2031, 3264, 3330
CJKglue	3, 3260
CJKmath	3
\CJKrmdefault	10
\CJKsfdefault	10
\CJKsout	17
CJKspace	3, 3317
\CJKsymbol	794, 826, 1045
\CJKttdefault	10
\CJKkunderanyline	18
\CJKkunderanysymbol	18
\CJKkunderdblline	17
\CJKkunderdot	17
\CJKkunderline	17
\CJKkunderwave	17
\CJKkout	17
CJLineBreak	6
CJStarter	388

clist commands:

\clist_const:Nn	398, 400, 402, 403, 410, 412, 417, 425, 427, 429, 439, 441, 442, 451, 452, 461, 511, 518, 520, 522
\clist_gclear:N	600, 601, 602, 603, 617, 618, 619, 620, 621, 622, 623, 624
\clist_gconcat:NNN	538, 552
\clist_if_in:nnTF	2541, 2597, 2632, 2666, 2725, 2728, 2782, 2808, 2821
\clist_map_function:NN	436
\clist_map_function:nN	712
\clist_map_inline:Nn	540, 590
\clist_map_inline:nn	34, 39, 753, 764, 766, 816, 834, 988, 3204
\clist_new:N	55, 346, 358
\clist_set:Nn	533
CM	388

cs commands:

<code>\cs_end:</code>	308
<code>\cs_generate_variant:Nn</code>	552, 642, 655, 661, 2685
<code>\cs_gset_eq:NN</code>	131, 142, 3243
<code>\cs_gset_protected:Npe</code>	690
<code>\cs_gset_protected:Npn</code>	3153, 3166, 3180, 3194, 3209, 3221
<code>\cs_if_eq:NNTF</code>	83, 142, 3242
<code>\cs_if_exist:NTF</code>	45, 95, 3135, 3162, 3176, 3190, 3206, 3218
<code>\cs_if_exist_use:NTF</code>	1254
<code>\cs_new:Npn</code>	112, 117, 201, 271, 273, 280, 286, 297, 311, 362, 524, 643, 1281, 1283
<code>\cs_new_eq:NN</code>	267, 268, 269, 363, 726, 951, 960, 1021, 1185, 3151, 3208, 3220, 3254, 3314, 3399
<code>\cs_new_protected:Npe</code>	3255
<code>\cs_new_protected:Npn</code>	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, 531, 536, 553, 577, 587, 637, 648, 650, 656, 662, 675, 686, 688, 704, 718, 797, 841, 846, 852, 858, 860, 900, 914, 940, 947, 952, 961, 1008, 1010, 1016, 1047, 1064, 1076, 1086, 1129, 1141, 1146, 1152, 1172, 1178, 1186, 1201, 1225, 1259, 1266, 1276, 1285, 1290, 1334, 1360, 1389, 1409, 1430, 1526, 1532, 1564, 1569, 1586, 1613, 1644, 1672, 1682, 1716, 1737, 1753, 1762, 1784, 1793, 1796, 1823, 1868, 1889, 1901, 1931, 1951, 1981, 2135, 2151, 2157, 2181, 2224, 2234, 2249, 2259, 2282, 2326, 2346, 2352, 2427, 2449, 2511, 2531, 2553, 2583, 2589, 2595, 2601, 2630, 2687, 2693, 2699, 2707, 2713, 2723, 2740, 2780, 2819, 2829, 2841, 2854, 2867, 2878, 2889, 2891, 2928, 2956, 2975, 2993, 3009, 3054, 3083, 3100, 3131, 3142, 3148, 3160, 3174, 3188, 3202, 3231, 3240, 3338, 3386, 3392, 3394, 3400
<code>\cs_set:Npn</code>	276, 433, 1228, 1234
<code>\cs_set_eq:NN</code>	129, 134, 135, 136, 240, 3165, 3179, 3193, 3280, 3281, 3282, 3283, 3284, 3291, 3294, 3295, 3304, 3305, 3306, 3307, 3308
<code>\cs_set_protected:Npe</code>	241, 243
<code>\cs_set_protected:Npn</code>	3264, 3273, 3302, 3323, 3329
<code>\cs_to_str:N</code>	2893, 2961, 3146, 3152, 3156
<code>\cs_undefine:N</code>	137, 437

ctex commands:

<code>\ctex_after_end_preamble:n</code>	72
<code>\ctex_at_end_package:nn</code>	86
<code>\ctex_at_end_preamble:n</code>	71
<code>\ctex_disable_package:n</code>	23, 41, 44
<code>\ctex_if_format_at_least:nTF</code>	42
<code>\ctex_patch_cmd:Nnn</code>	23
<code>\ctex_replace_package:nn</code>	43

D

Default	369
<code>\defaultCJKfontfeatures</code>	10
depth	18
dim commands:	
<code>\dim_case:nn</code>	1154
<code>\dim_case:nnTF</code>	1104, 2186

<code>\dim_compare:nNnTF</code>	888, 891, 1088, 1910
<code>\dim_compare_p:nNn</code>	2384, 2390, 2394
<code>\dim_const:Nn</code>	1263
<code>\dim_eval:n</code>	203
<code>\dim_gset:Nn</code>	1263
<code>\dim_if_exist:NTF</code>	1262
<code>\dim_max:nn</code>	2292
<code>\dim_new:N</code>	52, 880, 1421, 1422, 1423, 1424
<code>\dim_set:Nn</code>	885, 887, 2237, 2239, 2251, 2253, 2286, 2288, 2290, 2307, 2309, 2311
<code>\dim_set_eq:NN</code>	2265, 2269
<code>\dim_use:N</code>	224
<code>\dim_zero:N</code>	2261, 2262
<code>\c_zero_dim</code>	892, 1916, 2016, 2243, 2277, 2297, 2385, 2391, 2395

E

else commands:

<code>\else:</code>	146, 256, 261, 309, 317, 1240, 1257
EmboldenFactor	4
<code>\ensuremath</code>	1463, 1499
EnvCS	4
EnvCS+	4
EnvCS-	4

exp commands:

<code>\exp_after:wN</code>	140, 235, 253, 254, 259, 260, 262, 263, 274, 302
<code>\exp_args:Nc</code>	344, 1278, 1702, 2125, 2144, 2356, 3151
<code>\exp_args:Ncc</code>	1230
<code>\exp_args:NcNc</code>	3144
<code>\exp_args:Ne</code>	1448, 1502, 1987, 2569, 2610, 2765, 2893
<code>\exp_args:NNNo</code>	219, 223, 226
<code>\exp_args:NNV</code>	3063
<code>\exp_args:NnV</code>	3111, 3116, 3121
<code>\exp_args:NNVV</code>	3075
<code>\exp_args:NV</code>	1445, 1454, 1458, 1462, 1477, 1498, 1848, 1862, 1923, 1945, 1972, 2338, 2445, 2506, 2549, 3056
<code>\exp_args:NVV</code>	2496
<code>\exp_last_unbraced:Ne</code>	114
<code>\exp_not:N</code>	85, 242, 244, 258, 695
<code>\exp_not:n</code>	242, 244, 414, 415, 454, 455, 456, 457, 458, 459, 653, 659, 673, 683, 692
<code>\exp_stop_f:</code>	145, 316
experiment/boundary-register	7
experiment/halfright-prebreakpenalty	7
experiment/punct-measure-fix	7

F

FallBack	11
<code>\fbox</code>	3247
fi commands:	
<code>\fi:</code>	146, 235, 264, 265, 309, 317, 1240, 1257
format	17

fp commands:

<code>\fp_compare:nNnTF</code>	372
<code>\fp_eval:n</code>	204
FullLeft	369
FullRight	369

G

gap	18
group commands:	
\group_align_safe_begin:	246, 3343
\group_align_safe_end:	242, 244, 3393
\group_begin:	215, 275, 1227, 1425, 3144
\c_group_begin_token	720, 2154
\group_end:	226, 303, 1250, 1428, 3144
\c_group_end_token	726, 2348

H

HalfLeft	388
HalfRight	388
HangulJamo	388
HangulJamoL	388
HangulJamoT	388
HangulJamoV	388

hbox commands:

\hbox:n	82
\hbox_set:Nn	2406, 2435
\hbox_set:Nw	216
\hbox_set_end:	219, 223
\hbox_unpack:N	2408, 2437
\hbox_unpack_drop:N	2515

height	18
--------------	----

hidden	17
--------------	----

hyp internal commands:

_hyp_target_raise:n	82
----------------------------	----

I**if commands:**

\if_catcode:w	257
\if_cs_exist:w	306
\if_dim:w	1253
\if_int_compare:w	232, 315, 1239
\if_meaning:w	251

\IfBooleanT	529
-------------------	-----

InlineEnv	4
-----------------	---

InlineEnv+	4
------------------	---

InlineEnv-	4
------------------	---

int commands:

\int_case:nnTF	2413
\int_compare:nNnTF	150, 163, 183, 375, 954, 956, 1362, 1365, 1615, 1618, 1646, 1649, 1739, 1764, 1798, 1870, 1984
\int_const:Nn	69, 357, 380, 386, 1236
\int_div_truncate:nn	206
\int_eval:n	281, 288, 298, 312
\int_gdecr:N	1734, 1864, 1897, 2367, 2508
\int_gincr:N	1261, 1268, 1684, 1983
\int_gset:Nn	211, 1274, 2409
\int_gset_eq:NN	992, 1025, 2788
\int_if_exist:NTF	341, 354
\int_incr:N	574, 584
\int_max:nn	564
\int_min:nn	563
\int_new:N	50, 209, 236, 237, 1273, 1310, 1311, 2371
\int_set:Nn	559, 563, 564, 569, 570, 580, 589

\int_set_eq:NN	560
\int_step_inline:nn	1368, 1652
\int_until_do:nNnn	26
\int_use:N	371, 1592, 1599, 1606, 1624, 1630, 1636, 1687, 1693, 1698, 1704, 1710, 1722, 1728, 1776, 1810, 1836, 1844, 1880, 1988, 1992, 1997, 2002, 2007, 2012, 2020, 2027, 2034, 2040, 2046, 2052, 2058, 2064, 2070, 2083, 2092, 2102, 2111, 2119, 2127, 2140, 2146, 2166, 2172, 2358, 2380, 2432, 2455, 2463, 2492, 2501, 2517, 2524, 2538, 2564, 2574, 2592, 2710, 2745, 2752, 2760, 2770

\int_value:w	207
--------------------	-----

\int_zero:N	723
-------------------	-----

\c_one_int	329, 375
------------------	----------

\c_zero_int	66, 331, 954, 956, 1363, 1366, 1616, 1619, 1647, 1650, 1740, 1765, 1799, 1871
-------------------	---

K

KaiMingPunct	5
--------------------	---

KaiMingPunct+	5
---------------------	---

KaiMingPunct-	5
---------------------	---

keys commands:

\keys_define:nn	321, 3260, 3269, 3317
-----------------------	-----------------------

\keys_set:nn	609, 611, 2938
--------------------	----------------

L

LatinPunct	6
------------------	---

LoadFandol	9
------------------	---

LocalConfig	3
-------------------	---

LongPunct	5
-----------------	---

LongPunct+	5
------------------	---

LongPunct-	5
------------------	---

M

\makexeCJKactive	324, 328
------------------------	----------

\makexeCJKinactive	325, 328
--------------------------	----------

Mapping	9
---------------	---

\mbox	3246
-------------	------

MiddlePunct	5
-------------------	---

MiddlePunct+	5
--------------------	---

MiddlePunct-	5
--------------------	---

mode commands:

\mode_if_horizontal:TF	1391, 1874, 1903
------------------------------	------------------

\mode_if_math:TF	1743, 1768, 1802
------------------------	------------------

\mode_leave_vertical:	1746, 1771, 1805
-----------------------------	------------------

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

<code>\newCJKfontfamily</code>	9
<code>\NewDocumentCommand</code> 45, 100, 328, 330, 526, 550, 593, 598, 615	
<code>NewLineCS</code>	4
<code>NewLineCS+</code>	4
<code>NewLineCS-</code>	4
<code>\newXeTeXintercharclass</code>	344
<code>NoBreakCS</code>	6
<code>NoBreakCS+</code>	6
<code>NoBreakCS-</code>	6
<code>NoBreakLongPunct</code>	5
<code>NoBreakLongPunct+</code>	5
<code>NoBreakLongPunct-</code>	5
<code>NormalSpace</code>	388
<code>\normalspacedchars</code>	14, 593

P

para commands:	
<code>\para_end:</code>	7
peek commands:	
<code>\peek_after:Nw</code>	247, 253
<code>\peek_meaning_remove:NTF</code>	993, 1026
<code>\l_peek_token</code>	251, 258,
999, 1032, 3346, 3349, 3354, 3360, 3363, 3366, 3372, 3375	
<code>PlainEquation</code>	4
<code>PoZheHao</code>	388
<code>PoZheHaoLigature</code>	5
prg commands:	
<code>\prg_do_nothing:</code>	
129, 131, 268, 269, 649, 695, 951, 1185, 3254	
<code>\prg_new_conditional:Npnn</code>	
23, 143, 304, 313, 881, 1237, 1251, 1553, 2213, 2895	
<code>\prg_new_protected_conditional:Npnn</code>	
1438, 1472, 1482, 1514, 1542, 2374, 2401	
<code>\prg_return_false:</code>	26, 146, 309, 317,
894, 896, 898, 1240, 1257, 1443, 1451, 1465, 1480, 1486,	
1505, 1508, 1522, 1549, 1560, 2220, 2399, 2421, 2423, 2902	
<code>\prg_return_true:</code>	
26, 146, 309, 317, 893, 1240, 1257, 1450,	
1456, 1460, 1464, 1479, 1489, 1492, 1500, 1504, 1518,	
1521, 1550, 1556, 1559, 2216, 2219, 2398, 2420, 2898, 2901	
prop commands:	
<code>\prop_gput:Nnn</code> . 2833, 2845, 2858, 2871, 2882, 2890, 3075	
<code>\prop_if_in:NnTF</code>	
2831, 2843, 2856, 2869, 2880, 2897, 2900, 3063	
<code>\prop_map_inline:Nn</code>	3086
<code>\prop_new:N</code>	1313, 1314, 1315
<code>PunctBoundWidth</code>	6
<code>PunctFamily</code>	5
<code>PunctStyle</code>	5
<code>PunctWidth</code>	6

Q

quark commands:	
<code>\q_stop</code>	274, 276, 280, 286, 297, 302

R

<code>\RequirePackage</code>	21, 22, 46, 47
reverse commands:	
<code>\reverse_if:N</code>	232
<code>RubberPunctSkip</code>	6

S

<code>\sbox~</code>	83
scan commands:	
<code>\scan_stop:</code>	25, 28,
53, 54, 216, 240, 320, 1393, 1435, 1830, 2074, 3360, 3399	
<code>\s_stop</code>	115, 117
<code>sep</code>	18
seq commands:	
<code>\seq_gpop:NN</code>	1755, 1786, 1825, 1891
<code>\seq_gpush:Nn</code>	1741, 1744,
1747, 1766, 1769, 1772, 1800, 1803, 1806, 1872, 1876, 1885	
<code>\seq_gput_right:Nn</code>	347, 348, 359, 706, 3095
<code>\seq_gset_eq:NN</code>	699
<code>\seq_gset_from_clist:Nn</code>	701
<code>\seq_map_inline:Nn</code>	729, 742, 755, 777, 807, 3133
<code>\seq_new:N</code>	337, 338, 698, 700, 703, 1312, 1316, 1324
<code>\seq_pop_left:NN</code>	3103, 3105
<code>\seq_set_split:Nnn</code>	3102
<code>\setCJKfallbackfamilyfont</code>	11
<code>\setCJKfamilyfont</code>	9
<code>\setCJKmainfont</code>	9
<code>\setCJKmathfont</code>	10
<code>\setCJKmonofont</code>	9
<code>\setCJKsansfont</code>	9
<code>skip</code>	17
skip commands:	
<code>\skip_const:Nn</code>	1270
<code>\skip_gset:Nn</code>	1270
<code>\skip_horizontal:N</code> . 859, 949, 966, 973, 974, 981, 983,	
1095, 1096, 1110, 1120, 1136, 1137, 1143, 1182, 1196,	
1211, 1214, 1217, 1218, 1223, 1581, 1960, 1966, 1974, 2342	
<code>\skip_horizontal:n</code>	2704
<code>\skip_if_eq:nTF</code>	
152, 161, 169, 186, 970, 1203, 2340, 2558, 2621	
<code>\skip_if_exist:NTF</code>	1269
<code>\skip_if_finite:NTF</code>	883, 1954
<code>\skip_new:N</code> . 54, 987, 1417, 1418, 1419, 1420, 3268, 3315	
<code>\skip_set:Nn</code>	220, 224,
227, 886, 2236, 2241, 2275, 2299, 2316, 2334, 2616, 2618	
<code>\skip_set_eq:NN</code>	
904, 963, 1066, 1190, 1573, 1953, 2161, 2336, 2557	
<code>\skip_use:N</code>	
220, 227, 1609, 2023, 2030, 2037, 2086, 2095, 2169	
<code>\c_zero_skip</code>	152, 161, 169, 186, 2341, 2622
<code>SlantFactor</code>	4
<code>\SplitArgument</code>	551
str commands:	
<code>\c_backslash_str</code>	3060, 3067, 3091, 3138
<code>\str_case:nn</code>	1395, 1814, 3011, 3107
<code>\str_case:nnTF</code>	2977, 2996

`\str_const:Nn` 370
`\str_if_eq:nnTF` 17, 542, 731, 735,
 744, 757, 779, 809, 1851, 2284, 2471, 2513, 2603, 2608,
 2634, 2647, 2649, 2668, 2731, 2733, 2735, 2789, 2796, 2802
 subtract 17
 symbol 17
 sys commands:
`\sys_if_engine_xetex:TF` 10

T

TeX and L^AT_EX 2_ε commands:

`\(` 21, 51
`\)` 21, 55
`\@beginDvi` 24, 91, 95, 98
`\@empty` 24
`\@ifnextchar` 8, 83
`\@ifpackagelater` 19
`\@ifframebox` 83, 3220, 3221
`\@imakebox` 83, 3208, 3209
`\@isavepicbox` 83
`\@pkgextension` 28
`\@textcolor` 83
`\[` 4
`\$` 86
`\addCJKfontfeatures` 10
`\AtBeginDocument` 8, 52
`\AtBeginDvi` 24
`\AtBeginShipout` 20, 24
`\baselineskip` 3
`\begin` 4, 19
`\beginGroup` 38
`\box` 51
`\caption` 83
`\catcode` 20, 27, 54
`\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, 83
`\color@endgroup` 3236
`\color@setgroup` 3236
`\colorbox` 21
`\cprotect` 20, 21
`\defaultCJKfontfeatures` 10
`\document` 24
`\dotfill` 45
`\edef` 86
`\emph` 21
`\end` 4, 19
`\endgroup` 38
`\ensuremath` 21, 51, 55
`\ExplSyntaxOff` 7
`\ExplSyntaxOn` 7, 54
`\familydefault` 10
`\fbox` 21, 81
`\fontdimen7` 42
`\footnote` 6, 20
`\footnotemark` 6
`\framebox` 21, 83
`\global` 8, 83
`\halign` 53
`\hbox` 19
`\hfil` 63
`\hfill` 62, 63
`\hrulefill` 45
`\hskip` 3, 42, 43, 63
`\Hy@SaveSpaceFactor` 82
`\hyper@anchorstart` 82
`\hyperref` 21, 22
`\icprotect` 21
`\input` 54
`\kern` 7, 42, 46
`\lastnodetype` 70
`\lastskip` 42
`\leaders` 17, 43, 45
`\long` 8
`\makeatletter` 7
`\makeatother` 7
`\makebox` 21, 51, 83
`\maxdimen` 14–16
`\mbox` 3, 8, 21, 51, 81
`\meaning` 27
`\newCJKfontfamily` 5, 9
`\nfss@text` 71
`\nobreak` 6
`\normalfont` 10
`\normalspacedchars` 14
`\null` 50, 51, 63
`\outer` 20, 21
`\par` 4, 7
`\parfillskip` 7
`\path` 3
`\protected` 86
`\ref` 21, 22
`\relax` 21, 85
`\rmfamily` 9, 10
`\savebox` 8, 83
`\sbox` 8, 51, 83
`\setbox` 8, 42, 50, 51, 81, 83

<code>\setCJKfallbackfamilyfont</code>	11	<code>\tex_global:D</code>	2153
<code>\setCJKfamilyfont</code>	5, 9	<code>\tex_glueshrink:D</code>	197, 892, 2240, 2254
<code>\setCJKmainfont</code>	2, 9	<code>\tex_gluestretch:D</code>	196, 2238, 2252
<code>\setCJKmathfont</code>	10	<code>\tex_hbox:D</code>	2154, 3233
<code>\setCJKmonofont</code>	9	<code>\tex_hskip:D</code>	3257, 3258
<code>\setCJKsansfont</code>	9	<code>\tex_iffontchar:D</code>	145
<code>\sffamily</code>	9, 10	<code>\tex_italiccorrection:D</code>	993, 995, 1026, 1028
<code>\shipout</code>	24	<code>\tex_kern:D</code>	1287, 1288, 2016
<code>\spacefactor</code>	26, 51, 82	<code>\tex_lastkern:D</code>	1088, 1104, 1154, 1255
<code>\textbf</code>	21	<code>\tex_lastnodetype:D</code>	42, 956, 1239, 2410
<code>\textcolor</code>	3, 21	<code>\tex_lastskip:D</code>	46, 220,
<code>\textit</code>	21		904, 963, 1066, 1190, 1256, 1573, 1953, 2086, 2161, 2557
<code>\textnormal</code>	10	<code>\tex_let:D</code>	3397
<code>\textrm</code>	9, 10	<code>\tex_nullfont:D</code>	142
<code>\textsf</code>	9, 10	<code>\tex_par:D</code>	7
<code>\texttt</code>	9, 10	<code>\tex_penalty:D</code>	66, 68, 1181
<code>\ttfamily</code>	9, 10	<code>\tex_romannumeral:D</code>	255
<code>\UL@start</code>	43	<code>\tex_setbox:D</code>	2153, 3233
<code>\UL@stop</code>	43	<code>\tex_spacefactor:D</code>	992, 1025, 2788
<code>\uline</code>	19	<code>\tex_spaceskip:D</code>	152, 158, 161, 184, 190, 196, 197
<code>\ULon</code>	19	<code>\tex_the:D</code>	140, 645
<code>\ULthickness</code>	18	<code>\tex_unkern:D</code>	1293
<code>\unskip</code>	7	<code>\tex_unskip:D</code>	907, 964, 1069, 1191, 1296, 1576, 1956, 2176
<code>\url</code>	8	<code>\tex_XeTeXcharclass:D</code>	272, 375, 583, 591, 596
<code>\usepackage</code>	54	<code>\tex_XeTeXdashbreakstate:D</code>	723
<code>\verb</code>	8	<code>\tex_XeTeXinterchartokenstate:D</code>	329, 331
<code>\verbatim@font</code>	8	<code>\tex_XeTeXinterchartoks:D</code>	639, 645, 693
<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>textformat</code>	17
<code>\xeCJK@first@begindvi</code>	91, 92	<code>thickness</code>	18
<code>\xeCJKCancelSubCJKBlock</code>	13	<code>tl commands:</code>	
<code>\xeCJKchar</code>	22	<code>\c_space_tl</code>	3314
<code>\xeCJKDeclareCharClass</code>	13	<code>\tl_clear:N</code>	2183, 2935, 2936
<code>\xeCJKDeclarePunctStyle</code>	5, 14	<code>\tl_const:Nn</code>	28, 97, 148, 707, 1427
<code>\xeCJKDeclareSubCJKBlock</code>	13	<code>\tl_gclear:N</code>	1298, 1853, 1990,
<code>\xeCJKEditPunctStyle</code>	14		1995, 2000, 2005, 2010, 2068, 2131, 2364, 2434, 2467, 2585
<code>\xeCJKfntefbox</code>	18	<code>\tl_gput_right:Nn</code>	80, 82, 84, 96
<code>\xeCJKfntefon</code>	19	<code>\tl_gset:Nn</code>	1279, 1370, 1373,
<code>\xeCJKnobreak</code>	6, 20		1589, 1603, 1621, 1627, 1633, 1656, 1678, 1708, 1718,
<code>\xeCJKOffVerbAddon</code>	20		1774, 1808, 1878, 2018, 2025, 2032, 2038, 2044, 2050,
<code>\xeCJKResetCharClass</code>	6		2056, 2062, 2080, 2089, 2100, 2117, 2138, 2164, 2170, 2805
<code>\xeCJKRestoreSubCJKBlock</code>	13	<code>\tl_gset_eq:NN</code>	1702, 2125, 2441
<code>\xeCJKsetkern</code>	14-16	<code>\tl_head:n</code>	54, 1449, 1503
<code>\xeCJKsetup</code>	2, 3, 14, 15, 17-19	<code>\tl_if_blank:nTF</code>	559
<code>\xeCJKsetwidth</code>	6, 14	<code>\tl_if_blank_p:n</code>	556, 557
<code>\xeCJKShipoutHook</code>	20	<code>\tl_if_empty:NNTF</code>	666, 670, 679, 1371, 1442, 1654, 1855,
<code>\xeCJKVerbAddon</code>	8, 20		1860, 1905, 1934, 1958, 2107, 2476, 2485, 2504, 2939, 2944
<code>\XeTeXdashbreakstate</code>	40	<code>\tl_if_empty:NNTF</code>	278, 284, 290, 298, 1485
<code>\XeTeXinterchartoks</code>	21, 26	<code>\tl_if_eq:NNTF</code>	54, 1183, 1548
tex commands:		<code>\tl_if_eq:NnTF</code>	1375, 1379, 1596,
<code>\tex_afterassignment:D</code>	3396		1658, 1662, 1726, 1758, 1789, 1828, 1894, 1942, 1964,
<code>\tex_font:D</code>			2076, 2098, 2109, 2439, 2544, 2605, 2636, 2639, 2651,
	140, 145, 154, 155, 156, 166, 173, 174, 179, 180, 191		2654, 2670, 2673, 2701, 2715, 2742, 2749, 2757, 3015, 3023
<code>\tex_fontdimen:D</code>	154, 155, 156, 166, 173, 174, 179, 180, 191	<code>\tl_if_exist:NNTF</code>	25, 1338, 1685

`\tl_if_head_eq_meaning:nNTF` 53, 54, 1454, 1458, 1462, 1488, 1491, 1498
`\tl_if_head_is_group:nTF` 1445, 1494
`\tl_if_head_is_N_type:nTF` 121
`\tl_if_noalue:nTF` 563, 564
`\tl_if_single:nTF` 119
`\tl_if_single_token:nTF` 2958
`\tl_map_inline:nn` 595
`\tl_new:N` 48, 49, 87, 88, 89, 110, 1275,
1320, 1321, 1322, 1323, 1329, 1330, 1331, 1332, 1340,
1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349,
1350, 1351, 1352, 1353, 1415, 1416, 1691, 1696, 2373, 2426
`\tl_put_right:Nn` 91, 109
`\tl_replace_all:Nnn` 54, 681, 1434
`\tl_reverse:N` 54, 55, 1476
`\tl_set:Nn` 664, 668,
677, 1432, 1496, 1545, 1546, 1674, 1832, 1840, 1852,
2188, 2189, 2190, 2191, 2192, 2195, 2199, 2200, 2208,
2329, 2330, 2377, 2429, 2440, 2451, 2459, 2473, 2474,
2480, 2481, 2488, 2533, 2726, 2730, 2732, 2734, 2736,
2764, 2775, 2777, 2937, 2961, 2964, 2970, 2979, 2980,
2981, 2982, 2983, 2988, 2998, 2999, 3000, 3039, 3049, 3070
`\tl_set_eq:NN` 1861, 2505
`\tl_tail:n` 54, 1497
`\tl_to_str:n` 2987, 3004
`\tl_trim_left_spaces:n` 1497
`\tl_trim_right_spaces:n` 54, 1547
`\tl_trim_spaces:n` 115, 122, 125, 1433
`\tl_use:N` 74,
76, 78, 105, 1675, 1720, 1834, 1842, 2453, 2461, 2490,
2535, 2561, 2571, 2612, 2617, 2619, 2690, 2696, 2719, 2767

tl internal commands:

`\__tl_tl_head:w` 53

token commands:

`\c_math_toggle_token` 3340
`\c_space_token` 251
`\token_if_cs:NTF` 2960
`\token_if_eq_meaning:NNTF` 3348, 3360, 3374
`\token_if_group_begin:NTF` 3354, 3363
`\token_if_group_end:NTF` 3366
`\token_if_letter:NTF` 567
`\token_if_macro:NTF` 3346, 3372
`\token_if_space:NTF` 999, 1032
`\token_to_meaning:N` 274, 302
`\TrimSpaces` 526

U

use commands:

`\use:N` 525, 708, 1256, 1282, 1284, 2703, 3156
`\use_i:nn` 863
`\use_none:n` 231, 234, 235
`\use_none:nn` 177, 194

V

`Verb` 8

W

`WidowPenalty` 4

X

`xCJKecglue` 3, 3269

xeCJK commands:

`\xeCJK_add_to_shipout:n` 108, 108
`\xeCJK_allow_break:` 65, 65
`\xeCJK_app_inter_class_toks:nnn` .. 656, 656, 661, 838
`\xeCJK_Boundary_and_Default:` 837, 841, 841
`\xeCJK_Boundary_and_NormalSp:` 1007, 1008, 1008
`\xeCJK_check_for_ecglue:` .. 844, 960, 3283, 3294, 3307
`\xeCJK_check_for_ecglue_normalsp:`
..... 1009, 1021, 3285, 3296, 3309
`\xeCJK_check_for_glue:` 1040, 1047, 1047
`\xeCJK_check_for_xglue:` . 1176, 1185, 3282, 3293, 3306
`\xeCJK_CJK_and_Boundary:w` 3337, 3338, 3338
`\xeCJK_class_group_begin:` 718, 718, 821, 1041
`\xeCJK_class_group_end:` 718, 726, 830, 3389, 3390
`\xeCJK_class_num:n`
. 524, 524, 545, 548, 589, 596, 639, 640, 645, 646, 694, 695
`\xeCJK_clear_Boundary_and_CJK_toks:`
..... 686, 686, 690, 692, 824, 1042
`\xeCJK_clear_inter_class_toks:nn`
..... 648, 648, 671, 769, 770, 771, 772, 773, 823
`\xeCJK_copy_inter_class_toks:nnnn` 662, 662, 732,
734, 736, 746, 747, 759, 760, 767, 781, 782, 785, 811, 812
`\xeCJK_cs_clear:N` 128, 128, 3292, 3293
`\xeCJK_cs_gclear:N` 128, 130
`\xeCJK_declare_char_class:nN` 531, 534,
536, 604, 605, 606, 607, 625, 626, 627, 628, 629, 630, 632
`\xeCJK_declare_char_class:nn` 528, 531, 531
`\xeCJK_declare_glue_node:n` 1266, 1305
`\xeCJK_declare_node:n`
1259, 1259, 1300, 1301, 1302, 1303, 1304, 1306, 1307, 1308
`\xeCJK_fallback_symbol:NN` 793, 825, 1044
`\xeCJK_font_gset_to_current:N` 139, 139
`\xeCJK_FullRight_and_CJStarter:` 792, 797
`\xeCJK_get_inter_class_toks:nn`
..... 643, 643, 653, 659, 665, 669, 678
`\xeCJK_glue_to_skip:nN`
..... 213, 213, 2015, 2024, 2031, 3265, 3274, 3303
`\xeCJK_glyph_if_exist:N` 143
`\xeCJK_glyph_if_exist:NTF` 143
`\xeCJK_glyph_if_exist_p:N` 143
`\xeCJK_if_CJK_class:N` 304
`\xeCJK_if_CJK_class:NTF` 304
`\xeCJK_if_CJK_class_p:N` 304
`\xeCJK_if_last_node:n` 1251
`\xeCJK_if_last_node:nTF` 862, 864,
867, 870, 916, 919, 922, 925, 965, 968, 1018, 1192, 1195,
1207, 1210, 1213, 1251, 1558, 2205, 2215, 2218, 2226, 2328
`\xeCJK_if_last_node:TF` 1921, 2184, 2263, 2267
`\xeCJK_if_last_node_p:n` 1251
`\xeCJK_if_last_punct:TF` 1174
`\xeCJK_if_package_loaded:n` 23

`\xeCJK_if_package_loaded:nTF` 23, 36
`\xeCJK_if_package_loaded_p:n` 23
`\xeCJK_if_same_class:NN` 313
`\xeCJK_if_same_class:NNTF` 313
`\xeCJK_if_same_class_p:NN` 313
`\xeCJK_int_until_do:nn` 229, 229, 571, 581
`\xeCJK_inter_class_toks:nnn`
 637, 637, 642, 649, 652, 658,
 673, 682, 790, 818, 828, 836, 990, 1006, 1023, 1037, 3337
`\xeCJK_make_boundary:` 319, 319
`\xeCJK_make_node:n`
 996, 1001, 1029, 1034, 1259, 1276, 2792, 2798, 2812
`\xeCJK_make_space_node:`
 1000, 1033, 3254, 3254, 3281, 3292, 3305
`\xeCJK_new_class:n` 339, 339, 383,
 384, 385, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397
`\xeCJK_no_break:` 65, 67, 787, 789, 800
`\xeCJK_peek_catcode_ignore_spaces:N` 238, 238, 3340
`\xeCJK_pre_inter_class_toks:nnn`
 650, 650, 655, 787, 788, 1004
`\xeCJK_remove_node:`
 865, 868, 871, 927, 942, 966, 973, 1019,
 1056, 1092, 1108, 1114, 1117, 1157, 1159, 1161, 1163,
 1165, 1193, 1196, 1208, 1211, 1214, 1290, 2207, 2228, 2231
`\xeCJK_replace_inter_class_toks:nnnn` 675, 675
`\xeCJK_reset_space_factor:` 209, 210, 212, 722
`\xeCJK_save_class:nn` 352, 352, 369, 373, 374, 377, 378, 379
`\xeCJK_select_font:` 822, 1043
`\xeCJK_set_char_class:nnn` 544, 548, 577, 577
`\xeCJK_space_glue:` 1404, 2016, 3291, 3314, 3324
`\xeCJK_space_or_xecglue:`
 1019, 1400, 3280, 3291, 3304, 3389
`\c_xeCJK_space_skip_tl` 148, 886, 972, 1205, 2335
`\xeCJK_swap_cs:NN` 132, 132
`\xeCJK_tl_remove_outer_braces:n` 112, 112, 123
`\xeCJK_token_value_charcode:N` 272, 273, 273
`\xeCJK_token_value_class:N` 271, 271, 307, 315, 316
`\xeCJK_widow_penalty:` 1118, 1161, 1402, 2650
xeCJK internal commands:
`\__xeCJK_add_special_punct:nn` 435
`\__xeCJK_after_end_preamble:n` 70, 83
`\g__xeCJK_after_end_preamble_hook_tl` 78, 84, 89
`\__xeCJK_after_preamble:n` 70, 81, 90
`\g__xeCJK_after_preamble_hook_tl` 74, 82, 88
`\l__xeCJK_aligni_tl` 1183
`\c__xeCJK_alignment_tl` 54, 1427, 1435
`\__xeCJK_at_end_preamble:n` 70, 79
`\g__xeCJK_at_end_preamble_hook_tl` 76, 80, 87
`\g__xeCJK_base_class_seq` 698
`\l__xeCJK_begin_int`
 236, 569, 571, 573, 574, 579, 581, 583, 584
`\g__xeCJK_boundary_active_seq`
 1312, 1741, 1744, 1747, 1756, 1766, 1769,
 1772, 1787, 1800, 1803, 1806, 1826, 1872, 1876, 1885, 1892
`\l__xeCJK_boundary_active_tl`
 ... 1332, 1757, 1758, 1788, 1789, 1827, 1828, 1893, 1894
`\__xeCJK_boundary_box_begin:` 1310, 1748, 2135
`\__xeCJK_boundary_box_end:n` 1310, 1759, 2346
`\__xeCJK_boundary_box_end_transparent:n`
 1310, 2486, 2511
`\__xeCJK_boundary_box_finish:nn` 1310, 2349, 2361, 2449
`\__xeCJK_boundary_box_open:N` 1310, 2144, 2151
`\__xeCJK_boundary_box_set_last_from_node:` ...
 1310, 2427, 2469
`\__xeCJK_boundary_capture_allocate:n`
 1310, 1334, 1356, 1357, 1358, 1987
`\__xeCJK_boundary_capture_begin:`
 51, 1310, 1773, 1807, 1877, 1981, 2137
`\__xeCJK_boundary_capture_class:n`
 820, 831, 843, 1039, 1310, 1360, 1411, 1816, 1817
`\g__xeCJK_boundary_capture_depth_int`
 954, 1310, 1363,
 1368, 1592, 1599, 1606, 1616, 1624, 1630, 1636, 1647,
 1652, 1777, 1811, 1837, 1845, 1864, 1881, 1897, 1983,
 1985, 1988, 1993, 1998, 2003, 2008, 2013, 2021, 2028,
 2035, 2041, 2047, 2053, 2059, 2065, 2071, 2083, 2092,
 2103, 2112, 2120, 2128, 2141, 2147, 2167, 2173, 2359,
 2367, 2380, 2432, 2456, 2464, 2493, 2502, 2508, 2518,
 2525, 2538, 2564, 2574, 2592, 2710, 2745, 2752, 2760, 2770
`\__xeCJK_boundary_capture_emit_left:nn`
 1310, 1377, 1381, 1660, 1664, 1672
`\__xeCJK_boundary_capture_report_first:n`
 1310, 1644, 2482
`\__xeCJK_boundary_capture_resume:` . 1310, 1716, 3237
`\__xeCJK_boundary_capture_space:` . 1310, 2073, 2157
`\__xeCJK_boundary_capture_suspend:` . 1310, 1682, 3235
`\__xeCJK_boundary_CJK_and_math:` 3392
`\__xeCJK_boundary_emit_left:nn` 1310, 2496, 2589
`\__xeCJK_boundary_emit_left:nnn`
 1310, 1676, 2591, 2595, 2685
`\__xeCJK_boundary_emit_left_math_space:nnn` ...
 1310, 2598, 2601
`\__xeCJK_boundary_emit_left_normal:nnn`
 1310, 2599, 2630
`\__xeCJK_boundary_enable_source_space:n`
 1310, 2793, 2799, 2813, 2819
`\l__xeCJK_boundary_first_tl` 1330,
 1832, 1855, 1861, 2451, 2473, 2476, 2480, 2485, 2498, 2505
`\__xeCJK_boundary_group_end:n` . 3356, 3369, 3377, 3379
`\__xeCJK_boundary_group_math:w` 3355, 3364, 3394
`\__xeCJK_boundary_group_math_brace:` 3397, 3399
`\__xeCJK_boundary_group_math_peek:` 3396, 3400
`\__xeCJK_boundary_hmode_transparent_begin:` ...
 1310, 1868, 2861, 3168, 3182, 3196
`\__xeCJK_boundary_hmode_transparent_end:`
 1310, 1889, 2863, 3170, 3184, 3198
`\__xeCJK_boundary_identity:n` 3349, 3375
`\__xeCJK_boundary_if_capture_box_sized:` 1310, 2374
`\__xeCJK_boundary_if_capture_box_sized:TF` .. 2404
`\__xeCJK_boundary_if_capture_box_visible:` ...
 1310, 2401
`\__xeCJK_boundary_if_capture_box_visible:TF` . 2478

```

\_xeCJK_boundary_if_last_math_node:TF . 1310, 1553
\_xeCJK_boundary_if_last_math_space:TF . 1566, 1577
\_xeCJK_boundary_if_last_math_space:TF . 1310, 2213
\_xeCJK_boundary_if_last_math_space:TF . 873, 932, 977, 1078
\_xeCJK_boundary_if_math_edge:n . 1310, 1514
\_xeCJK_boundary_if_math_head:n . 54, 1310, 1438
\_xeCJK_boundary_if_math_head:nTF . 1448, 1517, 1528
\_xeCJK_boundary_if_math_tail:n . 54, 1310, 1472
\_xeCJK_boundary_if_math_tail:nTF . 1502, 1520, 1534
\_xeCJK_boundary_if_math_tail_reversed:n . 1483
\_xeCJK_boundary_if_math_tail_reversed:nTF . 1477
\_xeCJK_boundary_if_math_tail_space:n . 54, 1310, 1542
\_xeCJK_boundary_if_math_tail_space:nTF . 1536
\_xeCJK_boundary_if_registered:n . 1310, 2895
\_xeCJK_boundary_if_registered:nTF . 3056, 3088
\_xeCJK_boundary_inline_box_begin: 1310, 1737, 2848
\_xeCJK_boundary_inline_box_end:n . 1310, 1753, 2850
\_xeCJK_boundary_inline_last_box_begin: 1310, 1762, 2835, 3211, 3223
\_xeCJK_boundary_inline_last_box_end:n . 1310, 1784, 2837, 3215, 3227
\_xeCJK_boundary_inline_stream_begin:n . 1310, 1793, 2884
\_xeCJK_boundary_inline_stream_begin:nn . 1310, 1794, 1796
\_xeCJK_boundary_inline_stream_end:n . 1310, 1823, 2886
\_xeCJK_boundary_last_box_end:n . 1310, 1790, 2352
\l\_xeCJK_boundary_last_tl . 1331, 1840, 1852, 1860, 1861, 1862, 2459, 2474, 2481, 2504, 2505, 2506, 2726, 2730, 2732, 2734, 2736, 2764, 2775, 2777
\l\_xeCJK_boundary_math_actual_skip . 1418, 2095, 2236, 2238, 2240, 2241, 2252, 2254, 2275, 2321
\_xeCJK_boundary_math_begin: . 1310, 1389, 1412
\_xeCJK_boundary_math_begin:n 1526, 3155, 3212, 3224
\_xeCJK_boundary_math_begin_visible: 1310, 1409, 1529
\l\_xeCJK_boundary_math_delta_skip . 1420, 2299, 2316, 2341, 2342, 2622, 2625
\l\_xeCJK_boundary_math_ecglue_dim . 1422, 2288, 2294, 2302, 2309, 2313, 2319
\l\_xeCJK_boundary_math_ecglue_skip . 1419, 2289, 2301, 2310, 2318, 2337, 2618
\_xeCJK_boundary_math_end:n . 55, 1310, 1532, 3157, 3214, 3226
\_xeCJK_boundary_math_end_confirm: 1310, 1567, 1588, 1613
\_xeCJK_boundary_math_end_confirm_node: 1310, 1538, 1564
\_xeCJK_boundary_math_end_confirm_space: 1310, 1537, 1569
\_xeCJK_boundary_math_end_confirm_space-visible: 1310, 1579, 1586
\_xeCJK_boundary_math_set:n . 53, 1430, 1441, 1475
\l\_xeCJK_boundary_math_shrink_dim . 1424, 2239, 2245, 2253, 2256, 2262, 2266, 2279
\l\_xeCJK_boundary_math_skip . 1417, 1573, 1574, 1581, 1609, 2287, 2308, 2334, 2616
\_xeCJK_boundary_math_space_consume: 874, 933, 980, 1310, 2224, 2331
\_xeCJK_boundary_math_space_decode: 1310, 2196, 2229, 2259
\l\_xeCJK_boundary_math_space_dim . 1421, 2286, 2295, 2307, 2314
\_xeCJK_boundary_math_space_encode: 1310, 2249, 2791
\_xeCJK_boundary_math_space_expire: 1310, 1943, 1968, 2583
\_xeCJK_boundary_math_space_set_actual:n . 1310, 2234, 2569, 2610, 2765
\_xeCJK_boundary_math_space_set_delta:n . 1310, 2282, 2338, 2620
\_xeCJK_boundary_math_space_to_CJK: 1081, 1167, 1169, 1310, 2326
\l\_xeCJK_boundary_math_stretch_dim . 1423, 2237, 2244, 2251, 2255, 2261, 2270, 2278
\l\_xeCJK_boundary_math_tl . 53, 1415, 1432, 1434, 1442, 1446, 1449, 1455, 1459, 1463, 1476, 1478, 1496, 1499, 1545, 1548
\l\_xeCJK_boundary_math_type_tl 1416, 2329, 2330, 2339
\l\_xeCJK_boundary_node_tl 1329, 1394, 1395, 1546, 1548, 1674, 1677, 1831, 1849, 1933, 1934, 1942, 1946, 1957, 1958, 1964, 1973, 2075, 2076, 2098, 2107, 2130, 2438, 2439, 2440, 2442, 2488, 2497, 2533, 2542, 2544, 2549
\_xeCJK_boundary_peek_math:w 3350, 3376
\_xeCJK_boundary_pop_node:N . 1310, 1394, 1831, 1933, 1957, 2075, 2181, 2438
\l\_xeCJK_boundary_post_box . 1328, 1909, 1912, 1913, 1914, 1918, 1938, 1941, 1961, 1967, 1971, 1978
\_xeCJK_boundary_post_transparent: 1310, 1901, 2874
\_xeCJK_boundary_post_transparent_relocate: 1310, 1917, 1931
\_xeCJK_boundary_post_transparent_relocate-glue: 62, 1310, 1937, 1951
\_xeCJK_boundary_prepare_onearg_box:N . 1310, 3142, 3246, 3247
\_xeCJK_boundary_prepare_onearg_box:NN 1310, 3148
\_xeCJK_boundary_prepare_sbox: 3240, 3253
\l\_xeCJK_boundary_probe_box . 2372, 2406, 2412, 2435, 2444
\g\_xeCJK_boundary_probe_last_tl . 2426, 2434, 2442, 2446
\l\_xeCJK_boundary_probe_name_tl . 2373, 2377, 2385, 2391, 2395, 2408, 2429, 2437
\g\_xeCJK_boundary_probe_type_int . 2371, 2409, 2413
\_xeCJK_boundary_register_box:nn . 1310, 2829, 3111, 3248, 3249
\_xeCJK_boundary_register_makeboxes: 1310, 3202, 3250
\_xeCJK_boundary_register_post_transparent:n . 1310, 2867, 3127, 3251

```

```

\_xeCJK_boundary_register_stream:nn 1310, 2878, 3121
\_xeCJK_boundary_register_transparent:n ....
    ..... 1310, 2854, 3125
\_xeCJK_boundary_register_wrapped_box:nn ...
    ..... 1310, 2841, 3116
\g_xeCJK_boundary_registered_prop 78, 1313, 2831,
    2833, 2843, 2846, 2856, 2859, 2869, 2872, 2880, 2882, 2897
\_xeCJK_boundary_replay_before: .....
    ..... 1310, 1857, 1896, 2366, 2528, 2531
\_xeCJK_boundary_replay_before_math_space: ..
    ..... 1310, 2545, 2553
\_xeCJK_boundary_replay_node:n .....
    ..... 1310, 1862, 1924, 1946, 1973, 2506, 2546, 2549, 2578, 2780
\_xeCJK_boundary_reserve:N .....
    ..... 1310, 2891, 3150, 3164, 3178, 3192
\_xeCJK_boundary_reserve:n .....
    ..... 1310, 2889, 2893, 3205, 3252
\_xeCJK_boundary_reserve_space: ..... 3351
\g_xeCJK_boundary_reserved_prop .. 1314, 2890, 2900
\_xeCJK_boundary_restore_space: .....
    ..... 1310, 2116, 2550, 2707
\_xeCJK_boundary_restore_space:n .....
    ..... 1310, 2642, 2657, 2676, 2680, 2682, 2709, 2713
\_xeCJK_boundary_sbox:Nn ..... 3231, 3242, 3243
\_xeCJK_boundary_set_last_from_node:n .....
    ..... 1310, 1849, 2446, 2723
\_xeCJK_boundary_set_math_tail_from_capture:
    ..... 1310, 1850, 2470, 2740
\g_xeCJK_boundary_suspend_depth_int .....
    ..... 1311, 1366, 1619, 1650, 1684, 1688, 1694,
    1699, 1705, 1711, 1723, 1729, 1734, 1740, 1765, 1799, 1871
\_xeCJK_boundary_use_ccglue:n 1310, 2658, 2660, 2693
\_xeCJK_boundary_use_ecglue:n .....
    ..... 1310, 2641, 2644, 2662, 2675, 2678, 2687
\_xeCJK_boundary_use_glue:nn .....
    ..... 1310, 2624, 2689, 2695, 2699, 2718
\_xeCJK_boundary_use_ulem_glue:n ..... 43
\g_xeCJK_boundary_user_applied_seq 1316, 3095, 3133
\l_xeCJK_boundary_user_invalid_bool .....
    ..... 1318, 2933, 2942, 2947,
    2949, 2951, 2965, 2971, 2989, 3005, 3018, 3027, 3037, 3047
\l_xeCJK_boundary_user_mode_given_bool .....
    ..... 1319, 2934, 2995, 3033, 3043
\l_xeCJK_boundary_user_mode_tl .....
    ..... 1322, 2937, 2998, 2999, 3000, 3015,
    3023, 3036, 3039, 3046, 3049, 3073, 3106, 3112, 3117, 3122
\l_xeCJK_boundary_user_name_tl ..... 1320,
    2935, 2939, 2961, 2964, 2970, 3057, 3060, 3064, 3067, 3077
\_xeCJK_boundary_user_register:n ..... 1310, 2928
\_xeCJK_boundary_user_register_apply: . 1310, 3083
\_xeCJK_boundary_user_register_apply:nn ....
    ..... 1310, 3094, 3100
\_xeCJK_boundary_user_register_check: . 1310, 3131
\g_xeCJK_boundary_user_register_closed_bool .
    ..... 1317, 2930, 3085

\_xeCJK_boundary_user_register_command:n ...
    ..... 1310, 2956
\_xeCJK_boundary_user_register_mode:n . 1310, 2993
\_xeCJK_boundary_user_register_store: .....
    ..... 1310, 2952, 3054
\_xeCJK_boundary_user_register_strategy:n ...
    ..... 1310, 2975
\_xeCJK_boundary_user_register_validate: ...
    ..... 1310, 2950, 3009
\g_xeCJK_boundary_user_registered_prop .....
    ..... 1315, 3064, 3076, 3086
\l_xeCJK_boundary_user_strategy_tl .....
    ..... 1321, 2936, 2944,
    2979, 2980, 2981, 2982, 2983, 2988, 3011, 3072, 3104, 3107
\l_xeCJK_boundary_user_value_seq .....
    ..... 1324, 3102, 3104, 3106
\l_xeCJK_boundary_user_value_tl .. 1323, 3070, 3078
\_xeCJK_boundary_wrap_transparent_noarg:NN ..
    ..... 1310, 3160
\_xeCJK_boundary_wrap_transparent_onearg:NN .
    ..... 1310, 3174
\_xeCJK_boundary_wrap_transparent_onearg_-
    braced:NN ..... 1310, 3188
\_xeCJK_ccglue_or_space: .....
    ..... 1114, 1159, 1193, 1208, 1225, 3323, 3329
\l_xeCJK_ccglue_skip ..... 3265, 3268
\_xeCJK_check_for_ecglue: ... 841, 850, 852, 960, 3294
\_xeCJK_check_for_ecglue_aux: ... 42, 841, 856, 860
\_xeCJK_check_for_ecglue_normalsp: .....
    ..... 1014, 1016, 1022, 3297
\_xeCJK_check_for_glue_auxi: ..... 1050, 1152
\_xeCJK_check_for_glue_auxii: 1059, 1144, 1150, 1172
\_xeCJK_check_for_glue_auxiii: ..... 1175, 1178
\_xeCJK_check_for_glue_skip: ..... 1053, 1064
\_xeCJK_check_for_glue_skip_consume_and_-
    fallback: ..... 1074, 1146
\_xeCJK_check_for_glue_skip_kern: ..... 1071, 1076
\_xeCJK_check_for_glue_skip_kern_normal: ...
    ..... 1084, 1086
\_xeCJK_check_for_glue_skip_nonkern: .. 1072, 1129
\_xeCJK_check_for_glue_skip_restore: .....
    ..... 1082, 1098, 1124, 1126, 1139, 1141
\_xeCJK_check_for_xecglue: ..... 846, 3283, 3307
\_xeCJK_check_for_xecglue_normalsp: 1010, 3286, 3310
\_xeCJK_check_for_xglue: ..... 1186, 3282, 3306
\_xeCJK_check_for_xglue_aux: ..... 1197, 1201
\_xeCJK_check_num_range:nnNN ..... 553, 553, 579
\c_xeCJK_CJ_chars_clist ..... 442, 632
\l_xeCJK_CJ_strict_bool ..... 631, 635
\_xeCJK_CJK_and_Boundary_relax:N ..... 3361
\_xeCJK_CJK_and_math: ..... 3341, 3386, 3393
\c_xeCJK_CJK_chars_clist ..... 461, 625
\g_xeCJK_CJK_class_seq ..... 698
\_xeCJK_CJK_class_tl:n ..... 307, 311, 708
\l_xeCJK_CJK_group_bool ..... 320, 721, 725
\g_xeCJK_CJK_range_clist ..... 617

```

\g__xeCJK_CJStarter_range_clist	624	\l__xeCJK_last_penalty_int	1181
\c__xeCJK_CL_chars_clist	417, 454	\g__xeCJK_last_punct_tl	799, 801
\c__xeCJK_class_begin_int	380, 386	\l__xeCJK_last_skip	904, 905,
__xeCJK_class_csname:n		949, 963, 971, 974, 979, 981, 983, 987, 1066, 1067, 1096,	
..... 341, 345, 349, 354, 357, 362, 362, 363, 525, 708		1110, 1120, 1137, 1143, 1182, 1190, 1204, 1218, 1223,	
\g__xeCJK_class_seq		1953, 1954, 1960, 1966, 1974, 2161, 2162, 2169, 2557, 2559	
..... 337, 347, 359, 699, 729, 742, 755, 777, 807		\l__xeCJK_latin_punct_bool	610, 614
\c__xeCJK_CM_chars_clist	511, 627	\c__xeCJK_left_tl	1183
\g__xeCJK_CM_range_clist	619	__xeCJK_make_node:N	1278, 1285, 2202, 2255, 2256
\l__xeCJK_ecglue_skip	859, 966, 973,	__xeCJK_make_space_node:	2804, 3255, 3281, 3305
1095, 1136, 1196, 1211, 1214, 1217, 2337, 3274, 3303, 3315		__xeCJK_msg_new:nn	56,
\l__xeCJK_end_int	237, 570, 571, 579, 581	56, 364, 2905, 2910, 2912, 2914, 2916, 2918, 2920, 2925	
__xeCJK_error:n	56, 58, 2931, 2963, 2969	__xeCJK_msg_new:nnn	57
__xeCJK_error:nn	56, 59, 342, 355, 2941, 2946,	\g__xeCJK_new_class_seq	337, 348
2986, 3003, 3017, 3025, 3035, 3045, 3059, 3066, 3090, 3137		\c__xeCJK_nobreak_penalty_int	68, 69
\c__xeCJK_EX_chars_clist	439, 456	__xeCJK_node:n	1088,
__xeCJK_font_gset_to_current_aux:NN	140, 141	1106, 1113, 1115, 1156, 1158, 1160, 1162, 1164, 1166,	
\c__xeCJK_FullLeft_chars_clist	412, 606	1168, 1281, 2188, 2189, 2190, 2191, 2192, 2193, 2198, 2200	
\g__xeCJK_FullLeft_range_clist	602	\g__xeCJK_node_int ..	1261, 1264, 1268, 1271, 1273, 1274
\c__xeCJK_FullRight_chars_clist	452, 607	\g__xeCJK_non_CJK_class_seq	698
\g__xeCJK_FullRight_range_clist	603	\c__xeCJK_NormalSpace_chars_clist	398, 626
__xeCJK_get_charcode:w	274, 280, 286, 297	\g__xeCJK_NormalSpace_range_clist	618
__xeCJK_glue_check_expire_stale: 841, 875, 952, 1058		\c__xeCJK_NS_chars_clist	425, 455
\g__xeCJK_glue_check_pending_bool		\c__xeCJK_OP_chars_clist	403, 414
..... 42, 46, 902, 929, 944, 945, 957, 1080, 1090,		__xeCJK_orig_iframebox:w	3218, 3220, 3225
1093, 1101, 1103, 1131, 1132, 1148, 1149, 1309, 1713,		__xeCJK_orig_imakebox:w	3206, 3208, 3213
1732, 1733, 1854, 2132, 2232, 2365, 2468, 2586, 2827, 3368		\c__xeCJK_package_ext_tl	25, 28
__xeCJK_glue_node:n	1283, 3257, 3258	__xeCJK_package_hook:nn	70, 85
\c__xeCJK_HalfLeft_chars_clist	398, 604	__xeCJK_peek_catcode_false:w	243, 263, 269
\g__xeCJK_HalfLeft_range_clist	600	__xeCJK_peek_catcode_ignore_spaces_branches:w	
\c__xeCJK_HalfRight_chars_clist	398, 605	 247, 249, 254	
\g__xeCJK_HalfRight_range_clist	601	__xeCJK_peek_catcode_true:w	241, 260, 268
\g__xeCJK_HangulJamo_range_clist	620	\l__xeCJK_peek_ignore_spaces_bool	
\c__xeCJK_HangulJamoL_chars_clist	518, 628	 27, 245, 252, 270, 3344, 3388	
\g__xeCJK_HangulJamoL_range_clist	621	\l__xeCJK_peek_search_token	240, 258, 267
\c__xeCJK_HangulJamoT_chars_clist	518, 630	\c__xeCJK_PO_chars_clist	451, 458
\g__xeCJK_HangulJamoT_range_clist	623	\l__xeCJK_pozhehao_ligature_bool	608, 613
\c__xeCJK_HangulJamoV_chars_clist	518, 629	\c__xeCJK_PR_chars_clist	410, 415
\g__xeCJK_HangulJamoV_range_clist	622	__xeCJK_punct_glue:NN	801
\c__xeCJK_hyphens_chars_clist	425, 436, 459	__xeCJK_punct_rule:NN	799
__xeCJK_if_last_glue:TF	46, 217, 848, 854,	__xeCJK_recover_ecglue_source_space: 841, 855, 900	
1012, 1052, 1188, 1227, 1295, 1571, 1936, 2078, 2159, 2555		__xeCJK_recover_ecglue_source_space_aux: ...	
__xeCJK_if_last_hlist:TF	1227, 1907, 2354	 841, 908, 914	
__xeCJK_if_last_kern:TF	1049, 1070, 1227, 1292	__xeCJK_recover_ecglue_source_space_fallback:	
__xeCJK_if_last_math:TF 1055, 1133, 1216, 1227, 1555		 841, 910, 912, 951	
__xeCJK_if_last_none:TF	1227	__xeCJK_recover_ecglue_source_space_restore:	
__xeCJK_if_last_penalty:TF	1227	 841, 928, 934, 947	
__xeCJK_info:nnn	56, 64	__xeCJK_recover_ecglue_source_space_success:	
__xeCJK_int_until_do:wn	229, 231, 234, 235	 841, 917, 920, 923, 940	
\l__xeCJK_interword_dim	880, 887, 889	__xeCJK_replace_space:	849, 961, 961, 1013
\c__xeCJK_IS_chars_clist	441, 457	\l__xeCJK_reserve_space_bool	
\c__xeCJK_iteration_marks_chars_clist	427	 1109, 1119, 2049, 3322, 3328, 3336	
\l__xeCJK_last_kern_dim	2186, 2202, 2266, 2270	\c__xeCJK_right_tl	799, 801
\g__xeCJK_last_node_tl	46, 1275, 1279, 1298,	__xeCJK_save_CJK_class:n	704, 717
1707, 1718, 1853, 1905, 1924, 2131, 2364, 2467, 2585, 2805		__xeCJK_select_font:	802
\l__xeCJK_last_penalty_bool	1180	__xeCJK_set_char_class_aux:Nnw	531, 544, 550

_xeCJK_set_char_class_eq:nn	587, 587	_xeCJK_use_ecglue_skip:	858, 865, 868, 943, 1056, 1163, 1165
\l_xeCJK_shipout_hook_bool	102, 104, 111	_xeCJK_warning:n	60
\l_xeCJK_shipout_hook_tl	105, 109, 110	_xeCJK_warning:nn	56, 61
_xeCJK_skip_if_interword:N	46, 63, 841, 881	_xeCJK_warning:nnn	62
_xeCJK_skip_if_interword:NTF	905, 979, 1067, 1574, 2162	_xeCJK_warning:nnnn	63
\g_xeCJK_space_factor_int	150, 163, 183, 204, 207, 209, 992, 1025, 2788	\l_xeCJK_xecglue_bool	1094, 1135, 2055, 2332, 3279, 3290, 3301, 3316
_xeCJK_space_skip_scale:nnn	165, 171, 184, 188, 201	\c__xeCJK_xetex_version_str	370, 372
_xeCJK_swap_cs_aux:w	134, 136, 137	xeCJKactive	3, 321
_xeCJK_tl_remove_outer_braces:w	115, 117	\xeCJKCancelSubCJKBlock	13
_xeCJK_tmp:nn	1228, 1243, 1244, 1245, 1246, 1247, 1248, 1249	\xeCJKDeclareCharClass	13, 526
_xeCJK_tmp:w	276, 302, 433, 436, 437	\xeCJKDeclarePunctStyle	14
_xeCJK_tmp_aux:NNn	1230, 1234	\xeCJKDeclareSubCJKBlock	13
\l_xeCJK_tmp_bool	53	\xeCJKEditPunctStyle	14
\l_xeCJK_tmp_box	51, 216, 224	\xeCJKfntefon	19
\l_xeCJK_tmp_clist	55, 533, 534	\xeCJKknobreak	20
\l_xeCJK_tmp_dim	52, 885, 889, 2290, 2303, 2311, 2320	\xeCJKOffVerbAddon	20
\l_xeCJK_tmp_int	50, 363, 580, 583, 589, 591	\xeCJKResetCharClass	14, 615, 636
\l_xeCJK_tmp_skip	54, 886, 887, 2017, 2023, 2024, 2030, 2031, 2037	\xeCJKResetPunctClass	14, 529, 598, 633
\l_xeCJK_tmp_tl	48, 664, 666, 668, 670, 673, 677, 679, 681, 683	\xeCJKRestoreSubCJKBlock	13
\l_xeCJK_tmptb_tl	49	\xeCJKsetkern	14
_xeCJK_ulem_glue:n	43	\xeCJKsetup	2
_xeCJK_update_clear_toks:n	688, 710	\xeCJKsetwidth	14
		\xeCJKShipoutHook	20, 90
		\xeCJKVerbAddon	20