

xeCJK 宏包

CTEX.ORG

2026/07/15 v3.10.3*

目录

第 1 节 简介	1	3.7 其他	18
第 2 节 基本用法	2	第 4 节 已知问题和兼容性	18
第 3 节 用户手册	2	4.1 CJK 文字与命令交互时的间距	19
3.1 宏包选项	2	4.2 \char 原语与 interchar 的冲突	19
3.2 字体设置与选择	7	4.3 hyperref 链接与 CJKecglue	20
3.3 CJK 分区字体设置	11	第 5 节 xeCJK 代码实现	20
3.4 设置 CJK 字符范围	12	版本历史	186
3.5 标点符号的处理	12	代码索引	191
3.6 xeCJKfntef 用法说明	15		

第 1 节 简介

xeCJK 是一个 Xe_{La}TeX 宏包, 用于排版中日韩(CJK)文字。主要功能:

1. 分别设置 CJK 和英文字体;
2. 自动忽略 CJK 文字间的空格而保留其他空格, 允许在非标点汉字和英文字母 (a – z, A – Z) 间断行;
3. 提供多种标点处理方式: 全角式、半角式、开明式、行末半角式和 CCT 式;
4. 自动调整中英文间空白。

xeCJK 使用了 Xe_{La}TeX 的一些最新特性, 需要 Xe_{La}TeX 0.9995.0 (2009/06/29) 以后的版本。xeCJK 依赖 L^AT_EX 3 项目的宏包套件 [l3kernel](#) 和 [l3packages](#)。xeCJK 还需要通过 [fontspec](#) 宏包来调用系统字体。xeCJK 会自动根据需要载入这些宏包。

xeCJK 的原始作者是孙文昌, 2009 年 5 月起宏包被收入 [ctex-kit](#) 项目进行维护, 目前主要维护者是刘海洋¹、李清²及黄晨成³。

*[ctex-kit rev. 7fbcffa9](#).

¹leoliu.pku@gmail.com

²sobenlee@gmail.com

³liamhuang0205@gmail.com

第 2 节 基本用法

与其他 $\text{\LaTeX}$ 宏包一样, 引入 `xeCJK` 宏包只要在导言区使用

```
\usepackage{xeCJK}
```

在引入 `xeCJK` 宏包之后, 只要设置 CJK 文字的字体, 就可以在文档中使用中日韩文字了。

可以在各种文档类中使用 `xeCJK` 宏包, 最简单的示例是:

例 1

```
\documentclass{article}
\usepackage{xeCJK}
\setCJKmainfont{SimSun}

\begin{document}
中文 \LaTeX 示例。
\end{document}
```

上述示例设置了中文字体 `SimSun` (宋体)。运行此示例要求系统安装了设置的字体, 源文件用 UTF-8 编码保存, 使用 $\text{\XeTeX}$ 编译。

`xeCJK` 只提供了字体和标点控制等基本 CJK 语言支持。对于中文文档, 可以使用更为高层的 `ctex` 宏包或文档类, 它将自动调用 `xeCJK` 并设置好中文字体, 同时提供了进一步的本地化支持。详细内容参看 `ctex` 宏包套件的说明。

`xeCJK` 提供了大量选项, 可以在宏包调用时作为宏包选项或用 `\xeCJKsetup` 命令进行设置, 详见 3.1 节。除了 `\setCJKmainfont` 命令, `xeCJK` 还提供了许多其他命令设置和选择中文字体, 详见 3.2 节。其他更详细的功能也都将在下面详细说明。在本文档所在的文件夹的 `example` 目录下面也有一些例子可以参考。

第 3 节 用户手册

3.1 宏包选项

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

```
\xeCJKsetup \xeCJKsetup {<key_1>=<val_1>, <key_2>=<val_2>, ...}
```

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

例 2

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

等价于

例 3

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

带有 $\star$ 或者 $\star$ 标记的选项或命令只能在导言区中使用, 其中 $\star$ 还表示这个选项或命令只影响随后定义的 CJK 字体。其余不带特殊标记的选项或命令, 如果没有特别说明, 则可以在导言区或正文中使用。**粗体**表示 `xeCJK` 的默认设置。

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

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

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

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

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

xeCJKactive `xeCJKactive = {\true|false}`

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

CJKspace `CJKspace = {\true|false}`

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

CJKmath ★ `CJKmath = {\true|false}`

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

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

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

CJKecglue `CJKecglue = {\glue}`

设置 CJK 文字与西文、CJK 文字与行内数学公式之间的间距, 默认值是一个空格。使用这个选项设置的 ⟨`glue`⟩ 最好也要有一定的弹性。请注意, 这里设置的 ⟨`glue`⟩ 只影响 `xeCJK` 根据需要自动添加的空白, 源文件中直接输入的 CJK 文字与西文之间的空格不受影响 (直接输出)。有时候 `xeCJK` 可能不能正确地调整间距, 需要手动加空格。

xCJKecglue `xCJKecglue = {\true|false|glue}`

缺省状态下, `xeCJK` 不对源文件中直接输入的 CJK 文字与西文之间的空格进行调整, 如果需要调整, 请使用这个选项。如果使用这个选项, 将使用 `CJKecglue` 替换源文件中直接输入的 CJK 文字与西文之间的空格。

CheckSingle `CheckSingle = {\true|false}`

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

WidowPenalty `WidowPenalty = {\penalty|10000}`

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

PlainEquation PlainEquation = $\langle \text{true} | \text{false} \rangle$

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

NewLineCS NewLineCS = { $\backslash\text{par}$ $\backslash[$ }

NewLineCS+
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。

EnvCS EnvCS = { $\backslash\text{begin}$ $\backslash\text{end}$ }

EnvCS+
EnvCS- 设置 $\text{\LaTeX}$ 环境开始和结束的控制序列, 以便 CheckSingle 选项能够正确识别。以上是 xeCJK 的初始设置。

InlineEnv InlineEnv = { $\langle \text{env}_1 \rangle$, $\langle \text{env}_2 \rangle$, $\langle \text{env}_3 \rangle$, ...}

InlineEnv+
InlineEnv- 在使用 CheckSingle 选项的时候, xeCJK 会将 CJK 文字后接着的 $\text{\LaTeX}$ 环境的开始 $\backslash\text{begin}\{\dots\}$ 和结束 $\backslash\text{end}\{\dots\}$ 视为断行的地方, 如果有某些特殊的 $\text{\LaTeX}$ 环境没有造成断行, 可以使用这个选项来声明它, 以便 CheckSingle 能正确识别。

AutoFallBack AutoFallBack = $\langle \text{true} | \text{false} \rangle$

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

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

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 = $\{\langle \text{数字} | 4 \rangle\}$

设置伪粗体的默认粗细程度。

SlantFactor ☆ SlantFactor = $\{\langle \text{数字} | 0.167 \rangle\}$

设置伪斜体的倾斜程度, 范围是 $[-0.999, 0.999]$ 。

PunctStyle PunctStyle = $\{\langle \text{quanjiao} | \text{banjiao} | \text{kaiming} | \text{hangmobanjiao} | \text{CCT} | \text{plain} | \dots \rangle\}$

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 $\backslash\text{xeCJKDeclarePunctStyle}$ 定义新的标点格式。

PunctFamily PunctFamily = $\{\langle \text{false} | \text{family} \rangle\}$

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。 $\langle \text{family} \rangle$ 需要使用随后说明的 $\backslash\text{setCJKfamilyfont}$ 或 $\backslash\text{newCJKfontfamily}$ 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+ * 设置开明 (kaiming) 标点处理格式时的句末点号, KaiMingPunct 后带的 + 与 - 分别表示从已有的开明句末点号中增加或减少标点。

LongPunct * LongPunct = {< ——— …… >}

LongPunct+ * 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+ * 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。

NoBreakLongPunct- * 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+ * 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

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

LatinPunct LatinPunct = {true|false}

New: 2026-07-05

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

例 5

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

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

CJLineBreak CJLineBreak = {<normal|strict>}

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 きゃ 和 きゃっ 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 = {⟨length⟩}

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

PunctBoundWidth ★ PunctBoundWidth = {⟨length⟩}

与以上选项类似, 但设置的是标点符号出现在行首/尾时的宽度。

AllowBreakBetweenPuncts AllowBreakBetweenPuncts = ⟨true|false⟩

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

RubberPunctSkip RubberPunctSkip = ⟨true|false|plus|minus⟩

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 = ⟨true|false⟩

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

NoBreakCS NoBreakCS = { \footnote \footnotemark \nobreak }

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 10 000), 防止半角右标点出现在行首。

此选项的效果类似于 **LuaTeX-jā** 中的 `prebreakpenalty` 机制, 但受限于 **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}
```

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

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

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

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

3.2 字体设置与选择

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

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

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

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

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

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

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

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

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

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

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

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

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

`\CJKfamily` `\CJKfamily {<family>}`
`\CJKfamily + {<family>}`
`\CJKfamily - {<family>}`

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

`\newCJKfontfamily` ★ `\newCJKfontfamily [<family>] \<font-switch> {<font name>}[<font features>]` 或
`\newCJKfontfamily [<family>] \<font-switch> [<font features>] {<font name>}`

声明新的 CJK 字体族 `<family>` 并指定字体, 并定义 `\<font-switch>`, 在文档中可以使用它来切换 CJK 字体族。可以不必指定 `<family>`, 这时候 `<family>` 将等于 `<font-switch>`。

事实上, `\newCJKfontfamily` 是 `\setCJKfamilyfont` 和 `\CJKfamily` 的合并。例如

例 6

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

等价于

例 7

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

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

在文档中定义新的 CJK 字体族, 并马上使用它。

`\defaultCJKfontfeatures` ★ `\defaultCJKfontfeatures {<font features>}`

全局设置 CJK 字体族的默认选项。例如, 使用

例 8

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

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

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

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

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

例 9

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

可以将文档中当前使用的 CJK 主分区字体放大为 1.1。

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

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

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

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

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

就可以将全文的 CJK 和西文默认字体都改为无衬线字体族。

`\setCJKmathfont` ★ `\setCJKmathfont <{font name}>[<font features>]` 或
`\setCJKmathfont [<font features>] <{font name}>`
 设置数学公式中的 CJK 字体族。如果使用了 CJKmath 选项, 但是没有使用 `\setCJKmathfont` 设置数学公式中的 CJK 字体, 那么将使用 `\CJKfamilydefault` 作为数学公式中的 CJK 字体。

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

设置 CJK 字体族 `<family>` 的备用字体。例如, 使用

例 10

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

可以将 SimSun-ExtB 作为 SimSun 的备用字体。

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

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

例 11

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

如果 FallBack 的值为空, 将设置的是备用字体。例如,

例 12

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

等价于

例 13

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

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

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

例 14

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

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

3.2.1 XeTeX 的字体名查找

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

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

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

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

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

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

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

例 15

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

来设置字体。

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

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

这样就把字体列表保存在文件 `zhfont.txt` 中⁴。这样列出的字体列表就比较简明易用, 如 Windows 下预装的中文字体:

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

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

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

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

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

3.3 CJK 分区字体设置

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

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

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

例 16

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

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

例 17

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

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

例如, 可以使用

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

例 18

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

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

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

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

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

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

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

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

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

3.4 设置 CJK 字符范围

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

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

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

用于恢复 `xeCJK` 对各个字符类别的初始化设置。

```
\xeCJKResetCharClass *
```

用于重置标点符号所属的字符类。

```
\xeCJKResetPunctClass *
```

```
\normalspacedchars {<char list>}
```

在 `<char list>` 中出现的字符两端不自动添加空格，初始设置是 `/、\、和 - (U+002D)`。

3.5 标点符号的处理

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

3.5.1 设置特定标点符号的宽度和间距

这里的设置可用于除 `plain` 以外的所有标点处理格式。

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

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

`<标点列表>` 可以是单个标点，也可以是多个标点。例如，

例 19

```
\xeCJKsetwidth{。?}{0.7em}
```

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

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

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

例 20

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

将设置冒号与左双引号之间的空白宽度为 0.3em。

3.5.2 定义标点符号处理格式

```
\xeCJKDeclarePunctStyle * \xeCJKDeclarePunctStyle {<style>} {<options>}
```

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

```
\xeCJKEditPunctStyle * \xeCJKEditPunctStyle {<style>} {<options>}
```

修改已有的标点符号处理格式。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

下面选项设置标点符号出现在行首或者行尾时的宽度或比例。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

例 21

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

```

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

```

就定义了一个名为 mine 的标点处理格式。可以在导言区通过

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

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

例 22

```
\xeCJKEditPunctStyle { hangmoban jiao } { enabled-global-setting = false }
```

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

3.6 xeCJKfntef 用法说明

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

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

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

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

```

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

```

```

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

```

南朝梁劉勰文心雕龍養氣

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

在汉字下加点，可以和上述下划线命令嵌套使用。例如

虚室生白, 吉祥止止
 虚室生白, 吉祥止止

```

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

```

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

```

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

```

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

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

hidden 设置本选项为 true, 将隐藏文本内容, 只画下划线。

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

textformat 设置下划线或点的正文的格式。例如:

```
1 \CJKKunderline[textformat=\color{blue}]{虚室生白, 吉祥止止}\\
2 \CJKKunderdot[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 `\CJKKunderwave` 或 `\CJKKunderdot` 的符号。

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

瞻彼阒者, 虚室生白, 吉祥止止

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

thickness 设置 `\CJKKunderline`、`\CJKKunderdblline` 和 `\CJKKsout` 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 `\CJKKunderdot` 可能会影响到行距, 可以设置本选项进行调整。如果不希望 `\CJKKunderdot` 影响行距, 可以将本选项设置为 `0pt`。

sep 设置 `\CJKKunderdot` 与 `\CJKKunderline`、`\CJKKunderdblline` 或 `\CJKKunderwave` 嵌套使用时, 点与线或者线与点的距离。

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

height 设置删除线 `\CJKKsout` 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 `\CJKKsout` 的厚度和颜色, 让它具有类似高亮的效果:

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

瞻彼阒者, 虚室生白, 吉祥止止

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

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

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

例如, 高亮效果也可以如下实现:

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

虛室生白，吉祥止止。

`\CJKunderansymbol \CJKunderansymbol [〈选项〉] {〈深度〉} {〈符号〉} {〈文本内容〉}`

`\xCJKintef` 将 ⟨符号⟩ 放进一个盒子 (`\xCJKfntefbox`) 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 {瞻彼閼者，虚室生白，\CJKunderline{吉祥止止}}

瞻彼閼者，虚室生白，吉祥止止。

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

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

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

虛室生白, 吉祥止止

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

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

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

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

瞻 彼 閼 者，
虛室生白，吉祥止止。

```

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

```

瞻彼閼者，
虛室生白，吉祥止止。

```

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

```

3.7 其他

`\xeCJKVerbAddon`
`\xeCJKOffVerbAddon`

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

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

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

`\xeCJKShipoutHook`

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

第 4 节 已知问题和兼容性

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

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

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

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

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

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

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

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

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

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

`xeCJK` 依赖 $\text{\XeTeX}$ 的 `\XeTeXinterchartoks` 机制, 与使用相同机制的宏包 (例如 `polyglossia` 和 `xesearch`) 可能会存在不同程度的冲突。`xeCJK` 虽然为此作了一些处理, 但与它们共同使用时应该小心。

4.1 CJK 文字与命令交互时的间距

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

前 中 后	% 直接排版
前 \foo{中} 后	% 用命令包裹

两种写法应产生相同的 CJK 间距。

目前 `xeCJK` 已确保以下命令在此场景下的间距行为正确:

- 字体切换命令: `\textbf`、`\textit`、`\emph` 等。这些命令仅切换字体, 对 `interchar` 机制透明。
- 颜色命令: `\textcolor`、`\colorbox` 等 (基于 $\text{\LaTeX} 2_{\epsilon}$ 颜色机制的命令)。
- `xeCJKfntef` 命令: `\CJKunderdot`、`\CJKunderline`、`\CJKsout`、`\CJKunderwave`、`\CJKunderdblline`、`\CJKxout`。
- 盒子命令: `\mbox`。
- `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

需要注意的是, `xeCJK` 的间距兼容是针对上述命令的内部行为逐一实现的, 并非通用的自动检测机制。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。已知的限制包括: 使用 $\text{\LaTeX} 3$ 颜色后端 (`l3color`) 的命令尚未覆盖。如果遇到间距异常, 欢迎到 [GitHub Issues](#) 提交兼容请求。

4.2 \char 原语与 interchar 的冲突

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

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

`xeCJK` 已为以下宏包的命令提供了自动兼容补丁:

- `mtpro2`: `\overbrace`、`\underbrace`
- `pifont`: `\Pifont` (U 编码字体切换)

如果遇到其他宏包的类似问题, 可以在导言区手动包装有冲突的命令:

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

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

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

4.3 `hyperref` 链接与 `CJKecglue`

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

```
图\ref{fig}图 % 等价于"图1图"
```

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

第 5 节 `xeCJK` 代码实现

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

5.1 运行环境检查

`xeCJK` 必须使用 $\text{Xe}\TeX$ 引擎的支持。

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

应该使用较新版本的 `expl3` 宏包。

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

`\ctex_disable_package:n` 由 `ctexhook` 提供。

```
21 \RequirePackage { ctexhook }
```

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
22 \prg_new_conditional:Npnn \xeCJK_if_package_loaded:n #1 { p , T , F , TF }
23 {
24   \tl_if_exist:CTF { ver@ #1 . \c__xeCJK_package_ext_tl }
```

```

25     { \prg_return_true: } { \prg_return_false: }
26   }
27 \tl_const:Nc \c__xeCJK_package_ext_tl { \@pkgextension }

```

下面这些 CJK 系列宏包不应该被使用。

```

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

```

将 CJKfntef 包替换为 xeCJKfntef 包。

```

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

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

```

5.2 内部工具

分配临时变量。

```

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

```

各种信息函数的缩略形式。

```

\__xeCJK_msg_new:nn  \__xeCJK_error:nn  \cs_new_protected:Npn \__xeCJK_msg_new:nn { \msg_new:nnn { xeCJK } }
\__xeCJK_error:ne   \cs_new_protected:Npn \__xeCJK_msg_new:nnn { \msg_new:nnnn { xeCJK } }
\__xeCJK_warning:ne \cs_new_protected:Npn \__xeCJK_error:n { \msg_error:nn { xeCJK } }
\__xeCJK_info:nee   \cs_new_protected:Npn \__xeCJK_error:ne { \msg_error:nne { xeCJK } }
\cs_new_protected:Npn \__xeCJK_warning:n { \msg_warning:nn { xeCJK } }
\cs_new_protected:Npn \__xeCJK_warning:ne { \msg_warning:nne { xeCJK } }
\cs_new_protected:Npn \__xeCJK_warning:nee { \msg_warning:nnee { xeCJK } }
\cs_new_protected:Npn \__xeCJK_warning:neee { \msg_warning:nneee { xeCJK } }
\cs_new_protected:Npn \__xeCJK_info:nee { \msg_info:nnee { xeCJK } }

\xeCJK_allow_break:  \cs_new_protected:Npn \xeCJK_allow_break:
\xeCJK_no_break:    { \tex_penalty:D \c_zero_int }
\cs_new_protected:Npn \xeCJK_no_break:
{ \tex_penalty:D \c__xeCJK_nobreak_penalty_int }
\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 \AtBeginDocument { \xeCJK@document@hook }
\__xeCJK_after_preamble:n  \ctex_at_end_preamble:n { \xeCJK@document@left@hook }
\__xeCJK_after_end_preamble:n \ctex_after_end_preamble:n { \xeCJK@document@right@hook }
\__xeCJK_package_hook:nn \cs_new_protected:Npn \xeCJK@document@hook
{ \tl_use:N \g__xeCJK_after_preamble_hook_tl }

```

```

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

```

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

```

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

```

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

```

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

```

111 \cs_new:Npn \xeCJK_tl_remove_outer_braces:n #1
112   {
113     \exp_last_unbraced:Ne
114     \__xeCJK_tl_remove_outer_braces:w { \tl_trim_spaces:n {#1} } \s_stop
115   }
116 \cs_new:Npn \__xeCJK_tl_remove_outer_braces:w #1 \s_stop
117   {
118     \tl_if_single:NTF {#1}
119     {
120       \tl_if_head_is_N_type:NTF {#1}
121       { \tl_trim_spaces:n }
122       { \xeCJK_tl_remove_outer_braces:n }
123     }
124     { \tl_trim_spaces:n }
125     {#1}
126   }

```

\xeCJK_cs_clear:N 让控制序列的意义为空。

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

\xeCJK_swap_cs:NN 交换 #1 和 #2 的意义。

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

\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。

```
138 \cs_new_protected:Npn \xeCJK_font_gset_to_current:N
139   { \exp_after:wN \__xeCJK_font_gset_to_current_aux:NN \tex_the:D \tex_font:D }
140 \cs_new_protected:Npn \__xeCJK_font_gset_to_current_aux:NN #1#2
141   { \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。

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

\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。

```
147 \tl_const:Nn \c_xeCJK_space_skip_tl
148   {
149     \int_compare:nNnTF \g__xeCJK_space_factor_int = { 1000 }
150     {
151       \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
152       {
153         \tex_fontdimen:D 2 ~ \tex_font:D
154         plus \tex_fontdimen:D 3 ~ \tex_font:D
155         minus \tex_fontdimen:D 4 ~ \tex_font:D
156       }
157       { \tex_spaceskip:D }
158     }
159     {
160       \skip_if_eq:nnTF \tex_spaceskip:D \c_zero_skip
161       {
162         \int_compare:nNnTF \g__xeCJK_space_factor_int < { 2000 }
163         {
164           \__xeCJK_space_skip_scale:nnn
165           { \tex_fontdimen:D 2 ~ \tex_font:D }
166         }
167         {
168           \skip_if_eq:nnTF \tex_xspaceskip:D \c_zero_skip
169           {
170             \__xeCJK_space_skip_scale:nnn
171             {
172               \tex_fontdimen:D 2 ~ \tex_font:D +
173               \tex_fontdimen:D 7 ~ \tex_font:D
174             }
175           }
176           { \tex_xspaceskip:D \use_none:nn }
177         }
178         { \tex_fontdimen:D 3 ~ \tex_font:D }
179         { \tex_fontdimen:D 4 ~ \tex_font:D }
180       }
181     }
182   }
```

```

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

```

\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。

```

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

```

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

```

\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
228 \cs_new_protected:Npn \xeCJK_int_until_do:nn #1#2
229 {
230   \__xeCJK_int_until_do:wn \use_none:n
231   { \reverse_if:N \if_int_compare:w #1#2 }
232 }
233 \cs_new_protected:Npn \__xeCJK_int_until_do:wn \use_none:n #1
234 { #1 \exp_after:wN \__xeCJK_int_until_do:wn \fi: \use_none:n {#1} }
235 \int_new:N \l__xeCJK_begin_int
236 \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。

```

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

```

270 \cs_new:Npn \xeCJK_token_value_class:N #1
271 { \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。

```

272 \cs_new:Npn \xeCJK_token_value_charcode:N #1
273 { \exp_after:wN \__xeCJK_get_charcode:w \token_to_meaning:N #1 \q_stop }
274 \group_begin:
275   \cs_set:Npn \__xeCJK_tmp:w #1 ~ #2 ~ #3#4#5 \q_stop
276   {
277     \tl_if_empty:nTF { #4#5 }
278     {
279       \cs_new:Npn \__xeCJK_get_charcode:w ##1 ~ ##2 ~ ##3 \q_stop
280         { \int_eval:n { `##3 } }
281     }
282     {
283       \tl_if_empty:nTF {#5}
284       {
285         \cs_new:Npn \__xeCJK_get_charcode:w ##1 ~ ##2 ~ ##3##4 \q_stop
286         {

```

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>

```

287             \int_eval:n
288             {
289                 \tl_if_empty:nTF { ##4 }
290                 { `##3 }
291                 { ( `##3 - "D800 ) * "400 + ( `##4 - "DC00 ) + "10000 }
292             }
293         }
294     }
295     {
296         \cs_new:Npn \__xeCJK_get_charcode:w ##1 ~ ##2 ~ ##3##4 \q_stop
297         { \int_eval:n { \tl_if_empty:nTF { ##4 } { `##3 } { "20000 } } }
298     }
299 }
300 }
301 \exp_after:wN \__xeCJK_tmp:w \token_to_meaning:N ~~~~~20000 { } \q_stop
302 \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:NNTF
303 \prg_new_conditional:Npnn \xeCJK_if_CJK_class:N #1 { p , T , F , TF }
304 {
305     \if_cs_exist:w
306     \__xeCJK_CJK_class_tl:n { \xeCJK_token_value_class:N #1 }
307     \cs_end:
308     \prg_return_true: \else: \prg_return_false: \fi:
309 }
310 \cs_new:Npn \__xeCJK_CJK_class_tl:n #1
311 { c__xeCJK_CJK_class_ \int_eval:n {#1} _tl }

```

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

```

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

```

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

```

318 \cs_new_protected:Npn \xeCJK_make_boundary:
319 { \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。

```

320 \keys_define:nn { xeCJK / options }
321 {
322     xeCJKactive .choice: ,
323     xeCJKactive / true .code:n = { \makexeCJKactive } ,
324     xeCJKactive / false .code:n = { \makexeCJKinactive } ,
325     xeCJKactive .default:n = { true }
326 }

```

```

\makexeCJKactive 327 \NewDocumentCommand \makexeCJKactive { }
\makexeCJKinactive 328 { \tex_XeTeXinterchartokenstate:D = \c_one_int }
329 \NewDocumentCommand \makexeCJKinactive { }
330 { \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。

```

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

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

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

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

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

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

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

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

`xeCJK` 需要以下字符类别用于字符输出。其中 `Default`、`CJK`、`FullLeft`、`FullRight`、`Boundary` 为 $\text{\XeTeX}$ 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 $\text{\XeTeX}$ 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 368 \xeCJK_save_class:nn { Default } { 0 }
CJK
FullLeft XeTeX 0.99994 将字符类总数扩大到 409610。
FullRight
Boundary 369 \str_const:Nc \c__xeCJK_xetex_version_str
370 { \int_use:N \tex_XeTeXversion:D \tex_XeTeXrevision:D }
371 \fp_compare:nNnTF { \c__xeCJK_xetex_version_str } > { 0.99993 }
372 { \xeCJK_save_class:nn { Boundary } { 4095 } }
373 { \xeCJK_save_class:nn { Boundary } { 255 } }

```

LaTeX 2_ε 2016/02/01 不再预设置 CJK 字符类。

```

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

```

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

```

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

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

```
\c__xeCJK_HalfRight_chars_clist
\c__xeCJK_NormalSpace_chars_clist
399 \clist_const:Nn \c__xeCJK_HalfLeft_chars_clist
400 { "28 , "5B , "60 , "7B , "2329 }
401 \clist_const:Nn \c__xeCJK_HalfRight_chars_clist
402 { "21 , "22 , "25 , "27 , "29 , "2C , "2E , "3A , "3B , "3F , "5D , "7D , "232A }
403 \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	ㄿ	U+FE39	ㄿ	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。

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

\c__xeCJK_PR_chars_clist Prefix Numeric (PR)

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

```
411 \clist_const:Nn \c__xeCJK_PR_chars_clist
412 { "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。

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

\c__xeCJK_CL_chars_clist Close Punctuation (CL)

U+00B7	·	U+2019	,	U+201D	”	U+2013	—	U+2014	—	U+2025	⋯	U+2026	⋯
U+2027	·	U+2E3A	⋈	U+3001	、	U+3002	。	U+3009	〉	U+300B	》	U+300D	」
U+300F	』	U+3011	】	U+3015	〕	U+3017	〕	U+3019	〕	U+301B	〕	U+301E	」
U+301F	ㄿ	U+FE11	、	U+FE12	。	U+FE18	⌒	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。

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

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。

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

\c__xeCJK_EX_chars_clist Exclamation/Interrogation (EX)

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

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

\c__xeCJK_IS_chars_clist Infix Numeric Separator (IS)

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

```
442 \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+1B12	☒	U+1B150->"1B152	☒->"1B152	U+1B155	☒	U+1B164->"1B167

```
443 \clist_const:Nn \c__xeCJK_CJ_chars_clist
444 {
445   "3041", "3043", "3045", "3047", "3049", "3063", "3083", "3085", "3087", "308E",
446   "3095", "3096", "30A1", "30A3", "30A5", "30A7", "30A9", "30C3", "30E3", "30E5",
447   "30E7", "30EE", "30F5", "30F6", "30FC", "31F0", "31F1", "31F2", "31F3", "31F4",
448   "31F5", "31F6", "31F7", "31F8", "31F9", "31FA", "31FB", "31FC", "31FD", "31FE",
449   "31FF", "FF67", "FF68", "FF69", "FF6A", "FF6B", "FF6C", "FF6D", "FF6E", "FF6F",
450   "FF70", "1B132", "1B150 -> "1B152", "1B155", "1B164 -> "1B167
451 }
```

\c__xeCJK_PO_chars_clist Postfix Numeric (PO)

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

```
452 \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。

```

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

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

```

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

- 闽南语、客家话阴去和阳去声调标记

```

464   "02EA -> "02EB ,
```

- CJK Radicals Supplement (中日韩部首补充)

```

465   "2E80 -> "2EFF ,
```

- Kangxi Radicals (康熙字典部首)

```

466   "2F00 -> "2FDF ,
```

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

```

467   "2FF0 -> "2FFF ,
```

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

```

468   "3000 -> "303F ,
```

- Hiragana (日文平假名)

```

469   "3040 -> "309F ,
```

- Katakana (日文片假名)

```

470   "30A0 -> "30FF ,
```

- Bopomofo (注音字母)

```

471   "3100 -> "312F ,
```

- Hangul Compatibility Jamo (谚文兼容字母)

```

472   "3130 -> "318F ,
```

- Kanbun (象形字注释标志)

```

473   "3190 -> "319F ,
```

- Bopomofo Extended (注音字母扩展)

```

474   "31A0 -> "31BF ,
```

- CJK Strokes (中日韩笔画)

```

475   "31C0 -> "31EF ,
```

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

```

476   "31F0 -> "31FF ,
```

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

```

477   "3200 -> "32FF ,
```

- CJK Compatibility (中日韩兼容)

```

478   "3300 -> "33FF ,
```

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

479 "3400 -> "4DBF ,

- Yijing Hexagrams Symbols (易经六十四卦符号)

480 "4DC0 -> "4DFF ,

- CJK Unified Ideographs (中日韩统一表意文字)

481 "4E00 -> "9FFF ,

- Yi Syllables (彝文音节)

482 "A000 -> "A48F ,

- Yi Radicals (彝文字根)

483 "A490 -> "A4CF ,

- Hangul Syllables (谚文音节)

484 "AC00 -> "D7AF ,

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

485 "F900 -> "FAFF ,

- Vertical Forms (竖排形式)

486 "FE10 -> "FE1F ,

- CJK Compatibility Forms (中日韩兼容形式)

487 "FE30 -> "FE4F ,

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

488 "FF00 -> "FFEF ,

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

489 "16FE0 -> "16FFF ,

- Tangut (西夏文)

490 "17000 -> "187FF ,

- Tangut Components (西夏文部首)

491 "18800 -> "18AFF ,

- Khitan Small Script (契丹小字)

492 "18B00 -> "18CFF ,

- Tangut Supplement (西夏文增补)

493 "18D00 -> "18D7F ,

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

494 "18D80 -> "18DFF ,

- Kana Extended-B (日文假名扩展 B)

495 "1AFF0 -> "1AFFF ,

- Kana Supplement (日文假名增补)

496 "1B000 -> "1B0FF ,

- Kana Extended-A (日文假名扩展 A)

497 "1B100 -> "1B12F ,

- Small Kana Extension (小型日文假名扩展)

498 "1B130 -> "1B16F ,

- Nushu (女书)

```
499      "1B170 -> "1B2FF ,
```

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

```
500      "1F200 -> "1F2FF ,
```

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

```
501      "20000 -> "2A6DF ,
```

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

```
502      "2A700 -> "2B73F ,
```

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

```
503      "2B740 -> "2B81F ,
```

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

```
504      "2B820 -> "2CEAF ,
```

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

```
505      "2CEB0 -> "2EBEF ,
```

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

```
506      "2EBF0 -> "2EE5F ,
```

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

```
507      "2F800 -> "2FA1F ,
```

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

```
508      "30000 -> "3134F ,
```

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

```
509      "31350 -> "323AF ,
```

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

```
510      "323B0 -> "3347F
```

```
511    }
```

\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

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

- Diacritics (音调符号)

```
514      "302A -> "302F ,
```

- 日文假名浊点

```
515      "3099 -> "309A ,
```

- Variation Selectors (异体字选择符)

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

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

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

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

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

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

```
\xeCJKDeclareCharClass 527 \NewDocumentCommand \xeCJKDeclareCharClass { s > { \TrimSpaces } m m }
528 {
529   \xeCJK_declare_char_class:nn {#2} {#3}
530   \IfBooleanT {#1} { \xeCJKResetPunctClass }
531 }
```

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

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

```
\__xeCJK_check_num_range:nnNN 554 \cs_new_protected:Npn \__xeCJK_check_num_range:nnNN #1#2#3#4
555 {
556   \bool_lazy_or:nnTF
557   { \tl_if_blank_p:n {#1} }
558   { \tl_if_blank_p:n {#2} }
559   {
560     \int_set:Nn #3 { \tl_if_blank:nTF {#1} {#2} {#1} }
561     \int_set_eq:NN #3 #4
562   }
563   {
564     \int_set:Nn #3 { \int_min:nn {#1} { \tl_if_no_value:nTF {#2} {#1} {#2} } }
565     \int_set:Nn #4 { \int_max:nn {#1} { \tl_if_no_value:nTF {#2} {#1} {#2} } }
566   }
567 }
```

```

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

```

`\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。

```

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

```

`\__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。

```

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

```

`\normalspacedchars` 声明前后不加间距的字符。

```

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

```

`\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。

```

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

```

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

```

616 \NewDocumentCommand \xeCJKResetCharClass { }
617 {

```

```

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

```

设置字符类别。

```
637 \xeCJKResetCharClass
```

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

```

638 \cs_new_protected:Npn \xeCJK_inter_class_toks:nnn #1#2#3
639 {
640   \tex_XeTeXinterchartoks:D \xeCJK_class_num:n {#1} ~
641   \xeCJK_class_num:n {#2} = {#3}
642 }
643 \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。

```

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

```

`\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。

```

649 \cs_new_protected:Npn \xeCJK_clear_inter_class_toks:nn #1#2
650 { \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。

```

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

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

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

将 #1 和 #2 之间出现的 #3 用 #4 替换。

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

清除边界与 CJK 文字、全角左右标点之间的内容。

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

保存宏包预先定义的字符类。

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

```

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

```

719 \cs_new_protected:Npn \xeCJK_class_group_begin:
720 {
721     \c_group_begin_token
722     \bool_set_true:N \l__xeCJK_CJK_group_bool
723     \xeCJK_reset_space_factor:
724     \int_zero:N \tex_XeTeXdashbreakstate:D
725 }
726 \bool_new:N \l__xeCJK_CJK_group_bool
727 \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。

```

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

```

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。

```

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

```

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。

```

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

```

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。

```

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

```

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)。

```

806 \AtEndOfPackage

```

```

807 {
808   \seq_map_inline:Nn \g__xeCJK_class_seq
809   {
810     \str_if_eq:nnF {#1} { PoZheHao }
811     {
812       \xeCJK_copy_inter_class_toks:nnnn { PoZheHao } {#1} { FullRight } {#1}
813       \xeCJK_copy_inter_class_toks:nnnn {#1} { PoZheHao } {#1} { FullRight }
814     }
815   }
816 }

817 \clist_map_inline:nn { Default , HalfLeft , HalfRight , NormalSpace }
818 {
819   \xeCJK_inter_class_toks:nnn {#1} { CJK }
820   {
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} { \xeCJK_class_group_end: }
829 }

830 \clist_map_inline:nn { Default , HalfLeft }
831 {
832   \xeCJK_inter_class_toks:nnn { Boundary } {#1}
833   { \xeCJK_Boundary_and_Default: }
834   \xeCJK_app_inter_class_toks:nnn { CJK } {#1}
835   { \CJKecglue }
836 }

\xeCJK_Boundary_and_Default: 837 \cs_new_protected:Npn \xeCJK_Boundary_and_Default:
838 { \xeCJK_check_for_ecglue: }
839 \cs_new_protected:Npn \__xeCJK_check_for_xecglue:
840 {
841   \__xeCJK_if_last_glue:TF
842   { \__xeCJK_replace_space: }
843   { \__xeCJK_check_for_ecglue: }
844 }
845 \cs_new_protected:Npn \__xeCJK_check_for_ecglue:
846 {
847   \xeCJK_if_last_node:nTF { CJK }
848   { \use_i:nn }
849   { \xeCJK_if_last_node:nTF { CJK-widow } }
850   { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
851   {
852     \xeCJK_if_last_node:nTF { CJK-space }
853     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
854     { }
855   }
856 }
857 \cs_new_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:

```

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

```

858 \cs_new_protected:Npn \__xeCJK_replace_space:
859 {
860   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
861   \tex_unskip:D
862   \xeCJK_if_last_node:nTF { CJK-space }
863   { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
864   {
865     \xeCJK_if_last_node:nTF { CJK }
866     {
867       \skip_if_eq:nnTF

```

```

868         { \l__xeCJK_last_skip }
869         { \c_xeCJK_space_skip_tl }
870         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
871         { \skip_horizontal:N \l__xeCJK_last_skip }
872     }
873     { \skip_horizontal:N \l__xeCJK_last_skip }
874 }
875 }
876 \skip_new:N \l__xeCJK_last_skip

877 \clist_map_inline:nn { Default , HalfRight }
878 {
879     \xeCJK_inter_class_toks:nnn {#1} { Boundary }
880     {
881         \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
882         \peek_meaning_remove:NTF \tex_italiccorrection:D
883         {
884             \tex_italiccorrection:D
885             { \xeCJK_make_node:n { default } }
886         }
887         {
888             \token_if_space:NTF \l_peek_token
889             { \xeCJK_make_space_node: }
890             { { \xeCJK_make_node:n { default } } }
891         }
892     }
893     \xeCJK_pre_inter_class_toks:nnn {#1} { CJK } { \CJKecglue }
894 }

895 \xeCJK_inter_class_toks:nnn { Boundary } { NormalSpace }
896 { \xeCJK_Boundary_and_NormalSp: }

\xeCJK_Boundary_and_NormalSp: 897 \cs_new_protected:Npn \xeCJK_Boundary_and_NormalSp:
898 { \xeCJK_check_for_ecglue_normalsp: }
899 \cs_new_protected:Npn \__xeCJK_check_for_xecglue_normalsp:
900 {
901     \__xeCJK_if_last_glue:TF
902     { \__xeCJK_replace_space: }
903     { \__xeCJK_check_for_ecglue_normalsp: }
904 }
905 \cs_new_protected:Npn \__xeCJK_check_for_ecglue_normalsp:
906 {
907     \xeCJK_if_last_node:nT { CJK-space }
908     { \xeCJK_remove_node: \xeCJK_space_or_xecglue: }
909 }
910 \cs_new_eq:NN \xeCJK_check_for_ecglue_normalsp:
911 \__xeCJK_check_for_ecglue_normalsp:

912 \xeCJK_inter_class_toks:nnn { NormalSpace } { Boundary }
913 {
914     \int_gset_eq:NN \g__xeCJK_space_factor_int \tex_spacefactor:D
915     \peek_meaning_remove:NTF \tex_italiccorrection:D
916     {
917         \tex_italiccorrection:D
918         { \xeCJK_make_node:n { normalspace } }
919     }
920     {
921         \token_if_space:NTF \l_peek_token
922         { \xeCJK_make_space_node: }
923         { { \xeCJK_make_node:n { normalspace } } }
924     }
925 }

926 \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
927 {
928     \xeCJK_check_for_glue:
929     \xeCJK_class_group_begin:
930     \xeCJK_clear_Boundary_and_CJK_toks:
931     \xeCJK_select_font:

```

```

932 \xeCJK_fallback_symbol:NN
933 \CJKsymbol
934 }

\xecjk_check_for_glue: 935 \cs_new_protected:Npn \xeCJK_check_for_glue:
936 {
937   \__xeCJK_if_last_kern:TF
938   { \__xeCJK_check_for_glue_auxi: }
939   {
940     \__xeCJK_if_last_glue:TF
941     { \__xeCJK_check_for_glue_skip: }
942     {
943       \__xeCJK_if_last_whatsit:TF
944       { \__xeCJK_recover_glue_whatsit: }
945       {
946         \__xeCJK_if_last_math:TF
947         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
948         { \__xeCJK_check_for_glue_auxii: }
949       }
950     }
951   }
952 }

```

`\xeCJKfntef` 的命令内部使用 `ulem` 的 `\hbox` 排版 CJK 文字, 导致 $\text{\XeTeX}$ 的 `interchar` class 在命令结束后不是 CJK 而是 Boundary。源码空格因此产生 `inter-word glue` (而非被吃掉), 叠在 CJK kern pair 标记上方。本函数通过 `\unskip` 探测 `glue` 下方的 kern pair 来解决。kern pair 的探测由 `\g_@@_glue_check_pending_bool` 门控: 仅在 `\xeCJKfntef` 组结束或颜色恢复 (`\reset@color`) 之后才启用, 以避免在 `space=true` 模式下误替换用户输入的空格。该标志分别由 `\@@_ulem_group_end:n` 和 `\reset@color` 补丁设置。

对于 `\mbox`、`\colorbox` 等产生 `hbox` 的命令, `hbox` 节点隔在 kern pair 和 `glue` 之间, `\lastkern` 无法穿透。此时检查 `\unskip` 后的最后节点是否为 `hlist`, 若是则通过全局 `\g_@@_last_node_tl` 判断前方内容类型。该变量由 `\xeCJK_make_node:n` 设置, 能穿透 `hbox/group` 边界保持 CJK 标记。`hlist` 分支本身不设门控: 其安全性依赖于 `\g_@@_last_node_tl` 仅在 CJK 输出路径中被设置, 非 CJK 上下文中该变量为空, `\@@_check_for_glue_skip_hlist_aux:` 会直接走 `fallback`。对于 `\colorbox` 等带颜色的 `hbox` 命令, `\reset@color` 在 `hbox` 之后插入 `color pop whatsit`, 遮蔽了 `hlist` 检测。此时通过专用标志 `\g_@@_reset_color_pending_bool` 门控 `whatsit` 回退路径: 仅当 `\reset@color` 已标记 `hlist` 上下文时才启用, 避免 `\write` 等无关 `whatsit` 的干扰。

`listings` 等宏包在 CJK 字符间放置 `fil` 级别的 `glue` 用于列对齐, 通过 `\skip_if_finite:nTF` 将其排除。标题格式中的 `\quad` 等有意放置的间距没有 `shrink` 分量, 通过检测 `\tex_glueshrink:D` 来排除。

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

```

953 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip:
954 {
955   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
956   \skip_if_finite:nTF { \l__xeCJK_last_skip }
957   {
958     \dim_compare:nNnTF
959     { \tex_glueshrink:D \l__xeCJK_last_skip } > \c_zero_dim
960     {
961       \tex_unskip:D
962       \__xeCJK_if_last_kern:TF
963       { \__xeCJK_check_for_glue_skip_kern: }
964       { \__xeCJK_check_for_glue_skip_nonkern: }
965     }
966     { \__xeCJK_check_for_glue_skip_consume_and_fallback: }
967   }
968   { \__xeCJK_check_for_glue_skip_consume_and_fallback: }

```

```

969 }
970 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_kern:
971 {
972   \bool_if:NTF \g__xeCJK_glue_check_pending_bool
973   {
974     \bool_gset_false:N \g__xeCJK_glue_check_pending_bool
975     \dim_case:nnF { \tex_lastkern:D }
976     {
977       { \__xeCJK_node:n { CJK } }
978       { \xeCJK_remove_node: \CJKglue }
979       { \__xeCJK_node:n { CJK-space } }
980       { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
981       { \__xeCJK_node:n { CJK-widow } }
982       { \xeCJK_remove_node: \xeCJK_widow_penalty: \CJKglue }
983     }
984     { \__xeCJK_check_for_glue_skip_restore: }
985   }
986   { \__xeCJK_check_for_glue_skip_restore: }
987 }
988 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_nonkern:
989 {
990   \bool_if:NT \g__xeCJK_glue_check_pending_bool
991   { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
992   \__xeCJK_if_last_hlist:TF
993   { \__xeCJK_check_for_glue_skip_hlist_aux: }
994   {
995     \bool_if:NTF \g__xeCJK_reset_color_pending_bool
996     {
997       \bool_gset_false:N \g__xeCJK_reset_color_pending_bool
998       \__xeCJK_if_last_whatsit:TF
999       { \__xeCJK_check_for_glue_skip_hlist_aux: }
1000       { \__xeCJK_check_for_glue_skip_restore: }
1001     }
1002     { \__xeCJK_check_for_glue_skip_restore: }
1003   }
1004 }
1005 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_restore:
1006 {
1007   \skip_horizontal:N \l__xeCJK_last_skip
1008   \__xeCJK_check_for_glue_auxii:
1009 }
1010 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_consume_and_fallback:
1011 {
1012   \bool_if:NT \g__xeCJK_glue_check_pending_bool
1013   { \bool_gset_false:N \g__xeCJK_glue_check_pending_bool }
1014   \__xeCJK_check_for_glue_auxii:
1015 }
1016 \cs_new_protected:Npn \__xeCJK_check_for_glue_skip_hlist_aux:
1017 {
1018   \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK }
1019   { \tl_gclear:N \g__xeCJK_last_node_tl \CJKglue }
1020   {
1021     \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK-space }
1022     { \tl_gclear:N \g__xeCJK_last_node_tl \__xeCJK_ccglue_or_space: }
1023     {
1024       \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK-widow }
1025       {
1026         \tl_gclear:N \g__xeCJK_last_node_tl
1027         \xeCJK_widow_penalty: \CJKglue
1028       }
1029       {
1030         \skip_horizontal:N \l__xeCJK_last_skip
1031         \__xeCJK_check_for_glue_auxii:
1032       }
1033     }
1034   }
1035 }

```

```

1036 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxi:
1037 {
1038   \dim_case:nn { \tex_lastkern:D }
1039   {
1040     { \__xeCJK_node:n { CJK } }
1041     { \xeCJK_remove_node: \CJKglue }
1042     { \__xeCJK_node:n { CJK-space } }
1043     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1044     { \__xeCJK_node:n { CJK-widow } }
1045     { \xeCJK_remove_node: \xeCJK_widow_penalty: \CJKglue }
1046     { \__xeCJK_node:n { default } }
1047     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1048     { \__xeCJK_node:n { hyperref-default } }
1049     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1050   }
1051 }
1052 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxii:
1053 {
1054   \xeCJK_if_last_punct:TF
1055   { \__xeCJK_check_for_glue_auxiii: }
1056   { \xeCJK_check_for_xglue: }
1057 }
1058 \cs_new_protected:Npn \__xeCJK_check_for_glue_auxiii:
1059 {
1060   \bool_if:NT \l__xeCJK_last_penalty_bool
1061   { \tex_penalty:D \l__xeCJK_last_penalty_int }
1062   \skip_horizontal:N \l__xeCJK_last_skip
1063   \tl_if_eq:NNF \l__xeCJK_aligni_tl \c__xeCJK_left_tl { \CJKglue }
1064 }
1065 \cs_new_eq:NN \xeCJK_check_for_xglue: \prg_do_nothing:
1066 \cs_new_protected:Npn \__xeCJK_check_for_xglue:
1067 {
1068   \__xeCJK_if_last_glue:T
1069   {
1070     \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1071     \tex_unskip:D
1072     \xeCJK_if_last_node:nTF { CJK-space }
1073     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1074     {
1075       \xeCJK_if_last_node:nTF { default-space }
1076       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1077       { \__xeCJK_check_for_xglue_aux: }
1078     }
1079   }
1080 }
1081 \cs_new_protected:Npn \__xeCJK_check_for_xglue_aux:
1082 {
1083   \skip_if_eq:nnTF
1084   { \l__xeCJK_last_skip }
1085   { \c__xeCJK_space_skip_tl }
1086   {
1087     \xeCJK_if_last_node:nTF { CJK }
1088     { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1089     {
1090       \xeCJK_if_last_node:nTF { default }
1091       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1092       {
1093         \xeCJK_if_last_node:nTF { hyperref-default }
1094         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1095         {
1096           \__xeCJK_if_last_math:TF
1097           { \skip_horizontal:N \l__xeCJK_ecglue_skip }
1098           { \skip_horizontal:N \l__xeCJK_last_skip }
1099         }
1100       }
1101     }
1102   }

```

```

1103     { \skip_horizontal:N \l__xeCJK_last_skip }
1104   }
1105   \cs_new_protected:Npn \__xeCJK_ccglue_or_space:
1106   { \CJKglue }

```

$\backslash_xeCJK_if_last_none:TF$ 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
1107 \group_begin:
1108 \cs_set:Npn \__xeCJK_tmp:nn #1
1109 {
1110   \exp_args:Ncc \__xeCJK_tmp_aux:NNn
1111   { __xeCJK_if_last_ #1 : }
1112   { c__xeCJK_ #1 _node }
1113 }
1114 \cs_set:Npn \__xeCJK_tmp_aux:NNn #1#2#3
1115 {
1116   \int_const:Nn #2 {#3}
1117   \prg_new_conditional:Npnn #1 { T , F , TF }
1118   {
1119     \if_int_compare:w \tex_lastnodetype:D = #2
1120     \prg_return_true: \else: \prg_return_false: \fi:
1121   }
1122 }
1123 \__xeCJK_tmp:nn { none } { -1 }
1124 \__xeCJK_tmp:nn { hlist } { 1 }
1125 \__xeCJK_tmp:nn { math } { 10 }
1126 \__xeCJK_tmp:nn { glue } { 11 }
1127 \__xeCJK_tmp:nn { kern } { 12 }
1128 \__xeCJK_tmp:nn { penalty } { 13 }
1129 \__xeCJK_tmp:nn { whatsit } { 9 }
1130 \group_end:

\xeCJK_if_last_node_p:n 1131 \prg_new_conditional:Npnn \xeCJK_if_last_node:n #1 { p , T , F , TF }
\xeCJK_if_last_node:nTF 1132 {
1133   \if_dim:w
1134     \cs_if_exist_use:cTF { c__xeCJK_#1_node_dim }
1135     { = \tex_lastkern:D }
1136     { \use:c { c__xeCJK_#1_node_skip } = \tex_lastskip:D }
1137     \prg_return_true: \else: \prg_return_false: \fi:
1138   }

```

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

```

\__xeCJK_make_node:n
1139 \cs_new_protected:Npn \xeCJK_declare_node:n #1
1140 {
1141   \int_gincr:N \g__xeCJK_node_int
1142   \dim_if_exist:cTF { c__xeCJK_#1_node_dim }
1143   { \dim_gset:cn } { \dim_const:cn }
1144   { c__xeCJK_#1_node_dim } { \g__xeCJK_node_int sp }
1145 }
1146 \cs_new_protected:Npn \xeCJK_declare_glue_node:n #1
1147 {
1148   \int_gincr:N \g__xeCJK_node_int
1149   \skip_if_exist:cTF { c__xeCJK_#1_node_skip }
1150   { \skip_gset:cn } { \skip_const:cn }
1151   { c__xeCJK_#1_node_skip } { \g__xeCJK_node_int sp }
1152 }
1153 \int_new:N \g__xeCJK_node_int
1154 \int_gset:Nn \g__xeCJK_node_int { 10 }
1155 \tl_new:N \g__xeCJK_last_node_tl
1156 \cs_new_protected:Npn \xeCJK_make_node:n #1
1157 {
1158   \exp_args:Nc \__xeCJK_make_node:N { c__xeCJK_#1_node_dim }
1159   \tl_gset:Nn \g__xeCJK_last_node_tl {#1}
1160 }
1161 \cs_new:Npn \__xeCJK_node:n #1
1162 { \use:c { c__xeCJK_#1_node_dim } }

```

```

1163 \cs_new:Npn \__xeCJK_glue_node:n #1
1164 { \use:c { c__xeCJK_#1_node_skip } }
1165 \cs_new_protected:Npn \__xeCJK_make_node:N #1
1166 {
1167   \tex_kern:D - #1
1168   \tex_kern:D #1
1169 }
1170 \cs_new_protected:Npn \xeCJK_remove_node:
1171 {
1172   \__xeCJK_if_last_kern:TF
1173   { \tex_unkern:D \tex_unkern:D }
1174   {
1175     \__xeCJK_if_last_glue:T
1176     { \tex_unskip:D \tex_unskip:D }
1177   }
1178   \tl_gclear:N \g__xeCJK_last_node_tl
1179 }
1180 \cs_new_protected:Npn \__xeCJK_recover_ecglue_whatsit:
1181 {
1182   \__xeCJK_if_last_whatsit:T
1183   {
1184     \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK }
1185     { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1186     {
1187       \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK-widow }
1188       { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1189       {
1190         \tl_if_eq:NnT \g__xeCJK_last_node_tl { CJK-space }
1191         { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1192       }
1193     }
1194   }
1195 }
1196 \cs_new_protected:Npn \__xeCJK_recover_glue_whatsit:
1197 {
1198   \__xeCJK_if_last_whatsit:T
1199   {
1200     \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK }
1201     { \xeCJK_remove_node: \CJKglue }
1202     {
1203       \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK-space }
1204       { \xeCJK_remove_node: \__xeCJK_ccglue_or_space: }
1205       {
1206         \tl_if_eq:NnTF \g__xeCJK_last_node_tl { CJK-widow }
1207         { \xeCJK_remove_node: \xeCJK_widow_penalty: \CJKglue }
1208         {
1209           \tl_if_eq:NnT \g__xeCJK_last_node_tl { default }
1210           { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1211           \tl_if_eq:NnT \g__xeCJK_last_node_tl { hyperref-default }
1212           { \xeCJK_remove_node: \skip_horizontal:N \l__xeCJK_ecglue_skip }
1213         }
1214       }
1215     }
1216   }
1217 }
1218 \xeCJK_declare_node:n { CJK }
1219 \xeCJK_declare_node:n { CJK-space }
1220 \xeCJK_declare_node:n { default }
1221 \xeCJK_declare_node:n { CJK-widow }
1222 \xeCJK_declare_node:n { normalspace }
1223 \xeCJK_declare_glue_node:n { default-space }
1224 \xeCJK_declare_node:n { hyperref-default }

1225 \bool_new:N \g__xeCJK_glue_check_pending_bool
1226 \bool_new:N \g__xeCJK_reset_color_pending_bool

```

\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。

```

1227 \cs_new_eq:NN \xeCJK_make_space_node: \prg_do_nothing:
1228 \cs_new_protected:Npe \__xeCJK_make_space_node:
1229 {
1230   \tex_hskip:D - \__xeCJK_glue_node:n { default-space }
1231   \tex_hskip:D \__xeCJK_glue_node:n { default-space }
1232 }

```

CJKglue CJK 文字之间插入的 glue。

```

1233 \keys_define:nn { xeCJK / options }
1234 {
1235   CJKglue .code:n =
1236   {
1237     \cs_set_protected:Npn \CJKglue {#1}
1238     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ccglue_skip
1239   }
1240 }
1241 \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

```

1242 \keys_define:nn { xeCJK / options }
1243 {
1244   CJKecglue .code:n =
1245   {
1246     \cs_set_protected:Npn \CJKecglue {#1}
1247     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
1248   } ,
1249   xCJKecglue .choice: ,
1250   xCJKecglue / true .code:n =
1251   {
1252     \bool_set_true:N \l__xeCJK_xecglue_bool
1253     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
1254     \cs_set_eq:NN \xeCJK_make_space_node: \__xeCJK_make_space_node:
1255     \cs_set_eq:NN \xeCJK_check_for_xglue: \__xeCJK_check_for_xglue:
1256     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_xecglue:
1257     \cs_set_eq:NN
1258       \xeCJK_check_for_ecglue_normalsp:
1259       \__xeCJK_check_for_xecglue_normalsp:
1260   } ,
1261   xCJKecglue / false .code:n =
1262   {
1263     \bool_set_false:N \l__xeCJK_xecglue_bool
1264     \cs_set_eq:NN \xeCJK_space_or_xecglue: \xeCJK_space_glue:
1265     \xeCJK_cs_clear:N \xeCJK_make_space_node:
1266     \xeCJK_cs_clear:N \xeCJK_check_for_xglue:
1267     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_ecglue:
1268     \cs_set_eq:NN
1269       \xeCJK_check_for_ecglue_normalsp:
1270       \__xeCJK_check_for_ecglue_normalsp:
1271   } ,
1272   xCJKecglue / unknown .code:n =
1273   {
1274     \bool_set_true:N \l__xeCJK_xecglue_bool
1275     \cs_set_protected:Npn \CJKecglue {#1}
1276     \xeCJK_glue_to_skip:nN {#1} \l__xeCJK_ecglue_skip
1277     \cs_set_eq:NN \xeCJK_space_or_xecglue: \CJKecglue
1278     \cs_set_eq:NN \xeCJK_make_space_node: \__xeCJK_make_space_node:
1279     \cs_set_eq:NN \xeCJK_check_for_xglue: \__xeCJK_check_for_xglue:
1280     \cs_set_eq:NN \xeCJK_check_for_ecglue: \__xeCJK_check_for_xecglue:
1281     \cs_set_eq:NN
1282       \xeCJK_check_for_ecglue_normalsp:
1283       \__xeCJK_check_for_xecglue_normalsp:
1284   } ,
1285   xCJKecglue .default:n = { true }
1286 }

```

```

1287 \cs_new_eq:NN \xeCJK_space_glue: \c_space_tl
1288 \skip_new:N \l__xeCJK_ecglue_skip
1289 \bool_new:N \l__xeCJK_xecglue_bool

```

CJKspace 是否保留 CJK 文字间的空白, 默认不保留。

```

1290 \keys_define:nn { xeCJK / options }
1291 {
1292   CJKspace .choice: ,
1293   CJKspace / true .code:n =
1294   {
1295     \bool_set_true:N \l__xeCJK_reserve_space_bool
1296     \cs_set_protected:Npn \__xeCJK_ccglue_or_space:
1297       { \xeCJK_space_glue: }
1298   } ,
1299   CJKspace / false .code:n =
1300   {
1301     \bool_set_false:N \l__xeCJK_reserve_space_bool
1302     \cs_set_protected:Npn \__xeCJK_ccglue_or_space:
1303       { \CJKglue }
1304   } ,
1305   CJKspace .default:n = { true } ,
1306   space .meta:n = { CJKspace = true } ,
1307   nospace .meta:n = { CJKspace = false }
1308 }
1309 \bool_new:N \l__xeCJK_reserve_space_bool
1310 \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。

```

1311 \cs_new_protected:Npn \xeCJK_CJK_and_Boundary:w
1312 {
1313   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
1314   {
1315     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1316     { \xeCJK_class_group_end: \xeCJK_space_or_xecglue: }
1317     { \xeCJK_class_group_end: \CJKecglue }
1318   }
1319   {
1320     \group_align_safe_begin:
1321     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1322     {
1323       \token_if_macro:NTF \l_peek_token
1324       { \__xeCJK_boundary_reserve_space: }
1325       { \__xeCJK_boundary_group_end:n { CJK-space } }
1326     }
1327     {
1328       \token_if_eq_meaning:NNTF \l_peek_token \scan_stop:
1329       { \__xeCJK_CJK_and_Boundary_relax:N }
1330       {
1331         \token_if_group_end:NTF \l_peek_token
1332         {
1333           \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
1334           \__xeCJK_boundary_group_end:n { CJK }
1335         }
1336         { \__xeCJK_boundary_group_end:n { CJK } }
1337       }
1338     }
1339   }
1340 }
1341 \cs_new_protected:Npn \__xeCJK_boundary_reserve_space:
1342 { \__xeCJK_boundary_group_end:n { CJK-space } }

```

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

```

1343 \cs_new_protected:Npn \__xeCJK_CJK_and_Boundary_relax:N #1
1344 {
1345   \__xeCJK_boundary_group_end:n { CJK }
1346   \token_if_eq_meaning:NNTF #1 \scan_stop:
1347     {#1} { \cs_set_eq:NN #1 \scan_stop: #1 }
1348 }
1349 \cs_new_protected:Npn \__xeCJK_boundary_group_end:n #1
1350 {
1351   \group_align_safe_end:
1352   \xeCJK_class_group_end:
1353   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
1354   { \xeCJK_make_node:n {#1} }
1355 }

\xeCJK_ignore_spaces:w 1356 \cs_new_protected:Npn \xeCJK_ignore_spaces:w
1357 {
1358   \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
1359   {
1360     \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
1361       { \xeCJK_space_or_xecglue: } { \CJKecglue }
1362   }
1363   {
1364     \bool_if:NT \l__xeCJK_peek_ignore_spaces_bool
1365     {
1366       \dim_case:nn { \tex_lastkern:D }
1367       {
1368         { \__xeCJK_node:n { CJK } }
1369         { \xeCJK_remove_node: \xeCJK_make_node:n { CJK-space } }
1370         { \__xeCJK_node:n { default } }
1371         { \xeCJK_remove_node: \xeCJK_make_space_node: }
1372       }
1373       \group_align_safe_begin:
1374       \token_if_macro:NTF \l_peek_token
1375         { \__xeCJK_reserve_space_aux: }
1376         { \group_align_safe_end: }
1377     }
1378   }
1379 }
1380 \cs_new_protected:Npn \__xeCJK_reserve_space_aux:
1381 {
1382   \group_align_safe_end:
1383   \xeCJK_space_or_xecglue:
1384 }

1385 \xeCJK_inter_class_toks:nnn { CJK } { CJK }
1386 { \xeCJK_CJK_and_CJK:N }

\xeCJK_CJK_and_CJK:N 1387 \cs_new_protected:Npn \xeCJK_CJK_and_CJK:N
1388 {
1389   \CJKglue
1390   \xeCJK_fallback_symbol:NN
1391   \CJKsymbol
1392 }

1393 \xeCJK_inter_class_toks:nnn { FullLeft } { CJK }
1394 {
1395   \xeCJK_FullLeft_and_CJK:
1396   \xeCJK_fallback_symbol:NN
1397   \CJKsymbol
1398 }
1399 \xeCJK_inter_class_toks:nnn { FullRight } { CJK }
1400 {
1401   \xeCJK_FullRight_and_CJK:
1402   \xeCJK_fallback_symbol:NN
1403   \CJKsymbol
1404 }
1405 \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
1406 {

```

```

1407 \clist_map_inline:nn { FullLeft , FullRight }
1408 {
1409     \xeCJK_inter_class_toks:nne {#1} {##1}
1410     { \exp_not:c { xeCJK_Default_and_##1:nN } {#1} }
1411     \xeCJK_inter_class_toks:nne {##1} {#1}
1412     { \exp_not:c { xeCJK_##1_and_Default: } }
1413 }
1414 }

```

```

\l__xeCJK_halfright_nobreak_bool 1415 \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。

```

1416 \xeCJK_app_inter_class_toks:nnn { CJK } { HalfRight }
1417 {
1418     \bool_if:NT \l__xeCJK_halfright_nobreak_bool
1419     { \xeCJK_no_break: }
1420 }

```

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

```

1421 \xeCJK_inter_class_toks:nnn { FullRight } { HalfRight }
1422 {
1423     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1424     \xeCJK_class_group_end:
1425     \bool_if:NT \l__xeCJK_halfright_nobreak_bool
1426     { \xeCJK_no_break: }
1427     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1428 }

```

```

1429 \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
1430 { \xeCJK_Boundary_and_FullLeft:N }
1431 \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
1432 { \xeCJK_Boundary_and_FullRight:N }
1433 \xeCJK_inter_class_toks:nnn { FullLeft } { Boundary }
1434 { \xeCJK_FullLeft_and_Boundary: }
1435 \xeCJK_inter_class_toks:nnn { FullRight } { Boundary }
1436 { \xeCJK_FullRight_and_Boundary: }

```

```

\l__xeCJK_FullLeft_and_Boundary: 1437 \cs_new_protected:Npn \xeCJK_FullLeft_and_Boundary:
1438 {
1439     \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
1440     {
1441         \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1442         \xeCJK_class_group_end:
1443         \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
1444         \xeCJK_no_break:
1445         \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
1446         \__xeCJK_punct_boundary_guard:
1447     }
1448     {
1449         \xeCJK_class_group_end:
1450         \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
1451         \__xeCJK_nobreak_zero_glue:
1452     }
1453     \tex_ignorespaces:D
1454 }

```

```

\l__xeCJK_FullRight_and_Boundary: 1455 \cs_new_protected:Npn \xeCJK_FullRight_and_Boundary:
1456 {
1457     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1458     \xeCJK_class_group_end:
1459     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
1460     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1461     \__xeCJK_punct_boundary_guard:
1462     \tex_ignorespaces:D
1463 }

```

`\_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)。

```

1464 \bool_new:N \g__xeCJK_par_guard_bool
1465 \dim_new:N \g__xeCJK_par_guard_dim
1466 \cs_new_protected:Npn \_xeCJK_punct_boundary_guard:
1467 {
1468   \mode_if_inner:TF
1469   { \tex_penalty:D \c_zero_int }
1470   {
1471     \bool_if:NT \g__xeCJK_punct_measure_fix_bool
1472     {
1473       \bool_gset_true:N \g__xeCJK_par_guard_bool
1474       \dim_gset:Nn \g__xeCJK_par_guard_dim { \tex_lastskip:D }
1475     }
1476   }
1477 }

```

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。

```

1478 \clist_map_inline:nn
1479 {
1480   Default , HalfLeft , HalfRight , NormalSpace ,
1481   CJK , FullLeft , FullRight , CM , HangulJamo ,
1482   HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
1483 }
1484 {
1485   \xeCJK_pre_inter_class_toks:nnn { Boundary } {#1}
1486   { \bool_gset_false:N \g__xeCJK_par_guard_bool }
1487 }

```

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。

```

1488 \AddToHook { para/begin }
1489 { \bool_gset_false:N \g__xeCJK_par_guard_bool }
1490 \AddToHook { para/end }
1491 {
1492   \bool_if:NT \g__xeCJK_par_guard_bool
1493   {
1494     \tex_kern:D \g__xeCJK_par_guard_dim
1495     \bool_gset_false:N \g__xeCJK_par_guard_bool
1496   }
1497 }

```

`\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。

```

1498 \cs_new_protected:Npn \xeCJK_punct_node:N #1
1499 {
1500   \_xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
1501   \_xeCJK_make_node:N \l__xeCJK_tmp_dim
1502   \dim_set:Nn \l__xeCJK_tmp_dim { `#1 sp }
1503   \_xeCJK_make_node:N \l__xeCJK_tmp_dim
1504 }

```

`\_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”。

```

1505 \cs_new_protected:Npn \_xeCJK_punct_bound_unitization:NN #1#2
1506 {
1507   \dim_set:Nn #2

```

```

1508     {
1509         \dim_max:nn
1510         { \c_zero_dim }
1511         { \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_right_tl #1 }
1512     }
1513     \dim_compare:nNnF {#2} < { 1pt }
1514     { \dim_set:Nn #2 { -1pt * \dim_ratio:nn {#2} { \c_max_dim } } }
1515 }

```

```

\__xeCJK_punct_bound_kern:N 1516 \cs_new_protected:Npn \xeCJK_punct_bound_kern:N #1
\__xeCJK_punct_bound_kern:NN 1517 {
1518     \exp_after:wN \__xeCJK_punct_bound_kern:NN
1519     \g__xeCJK_last_punct_tl #1
1520 }
1521 \cs_new_protected:Npn \__xeCJK_punct_bound_kern:NN #1#2
1522 {
1523     \xeCJK_get_punct_bounds:NN \l__xeCJK_aligni_tl #1
1524     \xeCJK_get_punct_kerning:NN #1 #2
1525     \__xeCJK_punct_bound_unitization:NN #1 \l__xeCJK_tmp_dim
1526     \skip_set:Nn \l__xeCJK_punct_kern_skip
1527     { \__xeCJK_use_dim_or_skip:nnn { bound_kern } #1 #2 }
1528     \dim_compare:nNnF \l__xeCJK_tmp_dim = \l__xeCJK_last_bound_dim
1529     { \__xeCJK_punct_bound_kern_ratio:NN #1 #2 }
1530     \bool_if:NTF \l__xeCJK_last_penalty_bool
1531     {
1532         \tex_penalty:D \l__xeCJK_last_penalty_int
1533         \skip_horizontal:N
1534     }
1535     { \__xeCJK_punct_bound_kern_aux:NNN #1 #2 }
1536     \l__xeCJK_punct_kern_skip
1537 }
1538 \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。

```

1539 \cs_new_protected:Npn \__xeCJK_punct_bound_kern_ratio:NN #1#2
1540 {
1541     \dim_set:Nn \l__xeCJK_bound_dim
1542     { \__xeCJK_use_punct_dim:nnn { bound_width } #1 #2 }
1543     \dim_compare:nNnT \l__xeCJK_bound_dim > \c_zero_dim
1544     {
1545         \dim_compare:nNnF \l__xeCJK_last_bound_dim > \c_zero_dim
1546         {
1547             \dim_set:Nn \l__xeCJK_last_bound_dim
1548             {
1549                 - \l__xeCJK_last_bound_dim *
1550                 \dim_ratio:nn { \c_max_dim } { 1pt }
1551             }
1552         }
1553         \__xeCJK_punct_bound_kern_ratio_aux:N #2
1554     }
1555 }
1556 \cs_new_protected:Npn \__xeCJK_punct_bound_kern_ratio_aux:N #1
1557 {
1558     \skip_set:Nn \l__xeCJK_punct_kern_skip
1559     {
1560         \l__xeCJK_punct_kern_skip *
1561         \dim_ratio:nn
1562         {
1563             \l__xeCJK_last_bound_dim
1564             + \__xeCJK_use_punct_dim:nnn { bound } \c__xeCJK_left_tl #1
1565         }
1566         { \l__xeCJK_bound_dim }
1567     }
1568 }

```

```

\__xeCJK_nobreak_hskip:N 1569 \cs_new_protected:Npn \__xeCJK_nobreak_hskip:N
\__xeCJK_nobreak_hskip:n 1570 { \xeCJK_no_break: \skip_horizontal:N }
\__xeCJK_punct_bound_kern:N
\__xeCJK_punct_bound_breakable_kern:N

```

```

1571 \cs_new_protected:Npn \__xeCJK_nobreak_hskip:n
1572 { \xeCJK_no_break: \skip_horizontal:n }
1573 \cs_new_eq:NN \__xeCJK_punct_bound_kern:N \__xeCJK_nobreak_hskip:N
1574 \cs_new_protected:Npn \__xeCJK_punct_bound_breakable_kern:N
1575 {
1576   \tl_if_eq:NNTF \l__xeCJK_aligni_tl \c__xeCJK_right_tl
1577   {
1578     \tl_if_eq:NNTF \l__xeCJK_alignii_tl \c__xeCJK_left_tl
1579     { \skip_horizontal:N }
1580     { \__xeCJK_nobreak_hskip:N }
1581   }
1582   { \__xeCJK_nobreak_hskip:N }
1583 }
1584 \cs_new_protected:Npn \__xeCJK_punct_bound_kern_aux:NNN #1#2
1585 {
1586   \str_if_eq:nnTF {#1} {#2}
1587   { \__xeCJK_nobreak_hskip:N }
1588   {
1589     \__xeCJK_punct_if_long:NTF #1
1590     { \skip_horizontal:N }
1591     {
1592       \__xeCJK_punct_if_long:NTF #2
1593       { \skip_horizontal:N }
1594       { \__xeCJK_punct_bound_kern:N }
1595     }
1596   }
1597 }

1598 \clist_map_inline:nn { CJK , FullLeft , FullRight }
1599 {
1600   \clist_map_inline:nn { FullLeft , FullRight }
1601   {
1602     \xeCJK_inter_class_toks:nne {#1} {##1}
1603     { \exp_not:c { xeCJK_#1_and_##1:N } }
1604   }
1605 }

```

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

```

1606 \cs_new_protected:Npn \__xeCJK_punct_bound_rule:NN #1#2
1607 {
1608   \tex_vrule:D
1609   width - \__xeCJK_use_punct_dim:nnN { bound } #1 #2 ~
1610   depth \c_zero_dim
1611   height \c_zero_dim \scan_stop:
1612 }

```

__xeCJK_punct_rule:NN 用于减少标点符号的左/右空白。

```

1613 \cs_new_protected:Npn \__xeCJK_punct_rule:NN #1#2
1614 {
1615   \tex_vrule:D
1616   width \__xeCJK_use_punct_dim:nnN { rule } #1 #2 ~
1617   depth \c_zero_dim
1618   height \c_zero_dim \scan_stop:
1619 }

```

__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。

```

1620 \cs_new_protected:Npn \__xeCJK_punct_glue:NN #1#2
1621 { \__xeCJK_punct_hskip:n { \__xeCJK_use_dim_or_skip:nnN { glue } #1 #2 } }
1622 \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。

```

1623 \cs_new_protected:Npn \__xeCJK_punct_kern:NN #1#2
1624 {

```

```

1625 \str_if_eq:eeTF {#1} {#2}
1626 { \__xeCJK_punct_nobreak_kern:NN }
1627 {
1628   \bool_lazy_or:nnTF
1629   { \__xeCJK_punct_if_long_p:N #1 }
1630   { \__xeCJK_punct_if_long_p:N #2 }
1631   { \__xeCJK_punct_kern_break:NN #1 #2 }
1632   { \__xeCJK_punct_nobreak_kern:NN }
1633 }
1634 #1 #2
1635 }
1636 \cs_new_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_kern: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。

```

1637 \cs_new_protected:Npn \__xeCJK_punct_kern_break:NN #1#2
1638 {
1639   \exp_after:wN \__xeCJK_punct_if_right:NTF #1
1640   {
1641     \__xeCJK_punct_if_right:NTF #2
1642     {
1643       \bool_lazy_and:nnTF
1644       { \__xeCJK_punct_if_long_p:N #2 }
1645       { ! \__xeCJK_punct_if_nobreak_long_p:N #2 }
1646       { \__xeCJK_punct_breakable_kern:NN }
1647       { \__xeCJK_punct_nobreak_kern:NN }
1648     }
1649     { \__xeCJK_punct_breakable_kern:NN }
1650   }
1651   { \__xeCJK_punct_nobreak_kern:NN }
1652 }

```

```

\__xeCJK_punct_nobreak_kern:NN 1653 \cs_new_protected:Npn \__xeCJK_punct_nobreak_kern:NN #1#2
1654 { \__xeCJK_nobreak_hskip:n { \__xeCJK_use_dim_or_skip:nnn { kern } #1 #2 } }

```

```

\__xeCJK_punct_breakable_kern:NN 1655 \cs_new_protected:Npn \__xeCJK_punct_breakable_kern:NN #1#2
1656 {
1657   \exp_after:wN \__xeCJK_punct_if_right:NT #1
1658   { \__xeCJK_punct_rule:NN \c__xeCJK_right_tl #1 }
1659   \__xeCJK_punct_breakable_kern:n
1660   { \__xeCJK_use_dim_or_skip:nnn { bound_kern } #1 #2 }
1661   \__xeCJK_punct_if_right:NF #2
1662   { \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #2 }
1663 }
1664 \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。

```

1665 \tl_new:N \g__xeCJK_last_punct_tl

```

```

\xeCJK_FullLeft_and_CJK: 1666 \cs_new_protected:Npn \xeCJK_FullLeft_and_CJK:
1667 {
1668   \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
1669   {
1670     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1671     \xeCJK_no_break:
1672     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
1673   }
1674   { }
1675   \__xeCJK_select_font:
1676 }

```

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

```

1677 \cs_new_protected:Npn \xeCJK_FullLeft_and_Default:
1678 {
1679   \_\_xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
1680   {
1681     \_\_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1682     \xeCJK_class_group_end: \xeCJK_no_break:
1683     \_\_xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
1684   }
1685   {
1686     \xeCJK_class_group_end:
1687     \_\_xeCJK_nobreak_zero_glue:
1688   }
1689 }
1690 \cs_new_protected:Npn \_\_xeCJK_nobreak_zero_glue:
1691 {
1692   \tex_penalty:D \c__xeCJK_nobreak_penalty_int
1693   \skip_horizontal:N \c_zero_skip
1694 }
1695 \cs_new_protected:Npn \_\_xeCJK_zero_glue:
1696 { \skip_horizontal:N \c_zero_skip }

```

```

\xeCJK_FullRight_and_CJK: 1697 \cs_new_protected:Npn \xeCJK_FullRight_and_CJK:
1698 {
1699   \_\_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1700   \_\_xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1701   \_\_xeCJK_select_font:
1702   \CJKglue
1703 }

```

```

\xeCJK_FullRight_and_Default: 1704 \cs_new_protected:Npn \xeCJK_FullRight_and_Default:
1705 {
1706   \_\_xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1707   \xeCJK_class_group_end:
1708   \_\_xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
1709 }

```

```

\xeCJK_Default_and_FullLeft:nN 1710 \cs_new_protected:Npn \xeCJK_Default_and_FullLeft:nN #1#2
1711 {
1712   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #2
1713   \_\_xeCJK_Default_and_FullLeft_glue:N #2
1714   \xeCJK_class_group_begin:
1715   \xeCJK_select_punct_font:
1716   \xeCJK_clear_inter_class_toks:n {#1} { FullLeft }
1717   \xeCJK_clear_Boundary_and_CJK_toks:
1718   \tl_gset:Nn \g__xeCJK_last_punct_tl {#2}
1719   \_\_xeCJK_punct_rule:NN \c__xeCJK_left_tl #2
1720   \xeCJK_fallback_punct_symbol:NN
1721   \CJKpunctsymbol #2
1722 }
1723 \cs_new_protected:Npn \_\_xeCJK_Default_and_FullLeft_glue:N #1
1724 { \_\_xeCJK_punct_glue:NN \c__xeCJK_left_tl #1 }

```

```

\xeCJK_CJK_and_FullLeft:N 1725 \cs_new_protected:Npn \xeCJK_CJK_and_FullLeft:N #1
1726 {
1727   \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
1728   \_\_xeCJK_CJK_and_FullLeft_glue:N #1
1729   \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
1730   \_\_xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
1731   \_\_xeCJK_select_punct_font:
1732   \xeCJK_fallback_punct_symbol:NN
1733   \CJKpunctsymbol #1
1734 }
1735 \cs_new_protected:Npn \_\_xeCJK_CJK_and_FullLeft_glue:N #1
1736 {
1737   \CJKglue
1738   \_\_xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
1739 }

```

```

\XeCJK_Boundary_and_FullLeft:N 1740 \cs_new_protected:Npn \XeCJK_Boundary_and_FullLeft:N #1
1741 {
1742   \XeCJK_get_punct_bounds:NN \c__XeCJK_left_tl #1
1743   \__XeCJK_Boundary_and_FullLeft_glue:N #1
1744   \XeCJK_class_group_begin:
1745   \XeCJK_select_punct_font:
1746   \XeCJK_clear_Boundary_and_CJK_toks:
1747   \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
1748   \__XeCJK_punct_rule:NN \c__XeCJK_left_tl #1
1749   \XeCJK_fallback_punct_symbol:NN
1750   \CJKpunctsymbol #1
1751 }

```

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

```

1752 \cs_new_protected:Npn \__XeCJK_Boundary_and_FullLeft_glue:N #1
1753 {
1754   \tl_set_eq:NN \l__XeCJK_alignii_tl \c__XeCJK_left_tl
1755   \group_begin: \exp_args:NNc \group_end: \cs_if_exist_use:NTF
1756   { \__XeCJK_bound_type_ \int_use:N \etex_lastnodetype:D _glue:Nn }
1757   {#1}
1758   { \use:n }
1759   { \__XeCJK_punct_glue:NN \c__XeCJK_left_tl #1 }
1760 }
1761 \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。

```

1762 \cs_new_protected:cpn { \__XeCJK_bound_type_ -1 _glue:Nn } #1#2
1763 { \__XeCJK_zero_glue: }

```

__XeCJK_bound_type_1_glue:Nn 1 表示 hlist node, 在这里用来判断是否位于段首。基于正常情况下, TeX 会在段落开头插入宽度为 \parindent 的水平盒子用于缩进。

```

1764 \cs_new_protected:cpn { \__XeCJK_bound_type_ 1 _glue:Nn } #1
1765 {
1766   \int_do_while:nNnn \etex_lastnodetype:D = \c__XeCJK_hlist_node
1767   { \__XeCJK_bound_hbox_auxi: }
1768   \__XeCJK_if_last_none:TF
1769   {
1770     \dim_case:nnF { \box_wd:N \l__XeCJK_indent_box }
1771     {
1772       { \tex_parindent:D } { \__XeCJK_bound_hbox_auxii:nn }
1773       { \c_zero_dim } { \use_i:nn }
1774     }
1775     { \use:nn }
1776   }
1777   { \use:nn }
1778   { \hbox_unpack_drop:N \l__XeCJK_indent_box }
1779 }
1780 \cs_new_protected:Npn \__XeCJK_bound_hbox_auxi:
1781 {
1782   \box_set_to_last:N \l__XeCJK_tmp_box
1783   \hbox_set:Nn \l__XeCJK_indent_box
1784   {
1785     \box_use:N \l__XeCJK_tmp_box
1786     \hbox_unpack:N \l__XeCJK_indent_box
1787   }
1788 }
1789 \cs_new_protected:Npn \__XeCJK_bound_hbox_auxii:nn
1790 {
1791   \dim_compare:nNnTF
1792   { \box_ht:N \l__XeCJK_tmp_box } = \c_zero_dim
1793   { \use_i:nn }
1794   { \use:nn }
1795 }
1796 \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 L^AT_EX 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 L^AT_EX 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`。

```

1797 \cs_new_protected:cpn { \_xeCJK_bound_type_ 11 _glue:Nn } #1#2
1798 {
1799   \skip_if_finite:nTF { \tex_lastskip:D }
1800     { \_xeCJK_bound_glue_auxi:Nn #1 {#2} }
1801     { \_xeCJK_zero_glue: }
1802 }
1803 \cs_new_protected:Npn \_xeCJK_bound_glue_auxi:Nn #1#2
1804 {
1805   \_xeCJK_if_last_punct_glue:TF
1806     { \xeCJK_punct_bound_kern:N #1 }
1807     { \_xeCJK_bound_glue_auxii:n {#2} }
1808 }
1809 \cs_new_protected:Npn \_xeCJK_bound_glue_auxii:n #1
1810 {
1811   \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
1812   \skip_if_eq:nnTF { \l__xeCJK_last_skip } { 1sp }
1813     { \_xeCJK_zero_glue: }
1814     {
1815       \skip_if_eq:nnTF { \l__xeCJK_last_skip } { \labelsep }
1816       {
1817         \tex_unskip:D
1818         \_xeCJK_if_last_penalty:TF
1819           {
1820             \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
1821               { \skip_horizontal:N \l__xeCJK_last_skip }
1822               { \skip_horizontal:N \l__xeCJK_last_skip #1 }
1823           }
1824           { \skip_horizontal:N \l__xeCJK_last_skip #1 }
1825       }
1826       {#1}
1827     }
1828 }

```

`\_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`。

```

1829 \cs_new_protected:cpn { \_xeCJK_bound_type_ 12 _glue:Nn } #1#2
1830 {
1831   \xeCJK_if_last_node:nF { CJK }
1832     { \xeCJK_if_last_node:nF { CJK-space } { \use_none:nn } }
1833   \xeCJK_remove_node: \CJKglue
1834   #2
1835 }

```

`\_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 L^AT_EX 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。

```

1836 \cs_new_protected:cpn { \_xeCJK_bound_type_ 13 _glue:Nn } #1#2
1837 {
1838   \_xeCJK_if_last_punct_penalty:TF
1839     { \xeCJK_punct_bound_kern:N #1 }
1840     {
1841       \int_compare:nNnTF \tex_lastpenalty:D = \c_zero_int
1842       {
1843         \tex_unpenalty:D
1844         \_xeCJK_if_last_hlist:TF
1845           { \tex_penalty:D \c_zero_int }
1846           { \tex_penalty:D \c_zero_int #2 }
1847       }

```

```

1848         {#2}
1849     }
1850 }

```

```

\XeCJK_Default_and_FullRight:nN 1851 \cs_new_protected:Npn \XeCJK_Default_and_FullRight:nN #1#2
1852 {
1853     \XeCJK_get_punct_bounds:NN \c__XeCJK_right_tl #2
1854     \__XeCJK_Default_and_FullRight_glue:N #2
1855     \XeCJK_class_group_begin:
1856     \XeCJK_select_punct_font:
1857     \XeCJK_clear_inter_class_toks:nn {#1} { FullRight }
1858     \XeCJK_clear_Boundary_and_CJK_toks:
1859     \tl_gset:Nn \g__XeCJK_last_punct_tl {#2}
1860     \XeCJK_FullRight_symbol:N #2
1861 }

```

```

\XeCJK_Boundary_and_FullRight:N 1862 \cs_new_protected:Npn \XeCJK_Boundary_and_FullRight:N #1
1863 {
1864     \XeCJK_get_punct_bounds:NN \c__XeCJK_right_tl #1
1865     \XeCJK_if_last_punct:TF
1866     {
1867         \tl_set_eq:NN \l__XeCJK_alignii_tl \c__XeCJK_right_tl
1868         \XeCJK_punct_bound_kern:N
1869     }
1870     { \__XeCJK_Default_and_FullRight_glue:N }
1871     #1
1872     \XeCJK_class_group_begin:
1873     \XeCJK_select_punct_font:
1874     \XeCJK_clear_Boundary_and_CJK_toks:
1875     \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
1876     \XeCJK_FullRight_symbol:N #1
1877 }

```

```

\XeCJK_CJK_and_FullRight:N 1878 \cs_new_protected:Npn \XeCJK_CJK_and_FullRight:N #1
1879 {
1880     \XeCJK_get_punct_bounds:NN \c__XeCJK_right_tl #1
1881     \__XeCJK_CJK_and_FullRight_glue:N #1
1882     \tl_gset:Nn \g__XeCJK_last_punct_tl {#1}
1883     \__XeCJK_select_punct_font:
1884     \XeCJK_FullRight_symbol:N #1
1885 }

```

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

```

1886 \cs_new_protected:Npn \XeCJK_if_last_punct:TF
1887 {
1888     \bool_set_false:N \l__XeCJK_last_penalty_bool
1889     \__XeCJK_if_last_glue:TF
1890     { \__XeCJK_if_last_punct_glue:TF }
1891     {
1892         \__XeCJK_if_last_penalty:TF
1893         { \__XeCJK_if_last_punct_penalty:TF }
1894         { \use_ii:nn }
1895     }
1896 }
1897 \cs_new_protected:Npn \__XeCJK_if_last_punct_glue:TF
1898 {
1899     \prop_get:NoNTF \g__XeCJK_punct_skip_prop
1900     { \skip_use:N \tex_lastskip:D } \l__XeCJK_tmp_tl
1901     { \__XeCJK_if_last_punct_glue_auxi:TF }
1902     { \__XeCJK_if_last_punct_glue_auxii:TF }
1903 }
1904 \cs_new_protected:Npn \__XeCJK_if_last_punct_glue_auxi:TF
1905 {
1906     \skip_set_eq:NN \l__XeCJK_last_skip \tex_lastskip:D
1907     \tex_unskip:D
1908     \int_compare:nNnTF \tex_lastpenalty:D = \c__XeCJK_nobreak_penalty_int
1909     { \__XeCJK_if_last_punct_auxi:TF { \use_i:nn } }

```

```

1910     {
1911         \xeCJK_if_last_node:TF
1912         { \_xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
1913         { \use:n }
1914     }
1915     { \skip_horizontal:N \l__xeCJK_last_skip \use_ii:nn }
1916 }
1917 \cs_new_protected:Npn \_xeCJK_if_last_punct_glue_auxii:TF
1918 {
1919     \group_begin:
1920     \g__xeCJK_space_factor_int \tex_spacefactor:D
1921     \skip_if_eq:nnTF { \tex_lastskip:D } { \c_xeCJK_space_skip_tl }
1922     { \group_end: \_xeCJK_if_last_punct_glue_auxiii:TF }
1923     { \group_end: \use_ii:nn }
1924 }
1925 \cs_new_protected:Npn \_xeCJK_if_last_punct_glue_auxiii:TF
1926 {
1927     \skip_set_eq:NN \l__xeCJK_tmp_skip \tex_lastskip:D
1928     \tex_unskip:D
1929     \_xeCJK_if_last_glue:TF
1930     {
1931         \prop_get:NoNTF \g__xeCJK_punct_skip_prop
1932         { \skip_use:N \tex_lastskip:D } \l__xeCJK_tmp_tl
1933         { \_xeCJK_if_last_punct_glue_auxi:TF { \use_i:nn } }
1934         { \use:n }
1935     }
1936     { \use:n }
1937     { \skip_horizontal:N \l__xeCJK_tmp_skip \use_ii:nn }
1938 }
1939 \cs_new_protected:Npn \_xeCJK_if_last_punct_penalty:TF
1940 {
1941     \int_set_eq:NN \l__xeCJK_last_penalty_int \tex_lastpenalty:D
1942     \tex_unpenalty:D
1943     \bool_set_true:N \l__xeCJK_last_penalty_bool
1944     \_xeCJK_if_last_glue:TF
1945     { \_xeCJK_if_last_punct_glue:TF { \use_i:nn } }
1946     { \use:n }
1947     { \_xeCJK_last_punct_penalty_false:nn }
1948 }
1949 \cs_new_protected:Npn \_xeCJK_last_punct_penalty_false:nn #1#2
1950 {
1951     \bool_set_false:N \l__xeCJK_last_penalty_bool
1952     \tex_penalty:D \l__xeCJK_last_penalty_int
1953     #2
1954 }
1955 \cs_new_protected:Npn \_xeCJK_if_last_punct_auxi:TF
1956 {
1957     \tex_unpenalty:D
1958     \bool_if:NF \l__xeCJK_last_penalty_bool
1959     {
1960         \bool_set_true:N \l__xeCJK_last_penalty_bool
1961         \int_set_eq:NN \l__xeCJK_last_penalty_int \c__xeCJK_nobreak_penalty_int
1962     }
1963     \xeCJK_if_last_node:TF
1964     { \_xeCJK_if_last_punct_auxii:TF { \use_i:nn } }
1965     { \use:n }
1966     { \xeCJK_no_break: \use_ii:nn }
1967 }
1968 \cs_new_protected:Npn \_xeCJK_if_last_punct_auxii:TF
1969 {
1970     \dim_compare:nNnTF \l__xeCJK_last_kern_dim > \c_zero_dim
1971     { \_xeCJK_if_last_punct_auxiii:TF }
1972     { \_xeCJK_make_node:N \l__xeCJK_last_kern_dim \use_ii:nn }
1973 }
1974 \cs_new_protected:Npn \_xeCJK_if_last_punct_auxiii:TF
1975 {
1976     \int_case:nnTF { \tex_XeTeXcharclass:D \l__xeCJK_last_kern_dim }

```

```

1977 {
1978   { \xeCJK_class_num:n { FullRight } }
1979   { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_right_tl }
1980   { \xeCJK_class_num:n { FullLeft } }
1981   { \tl_set_eq:NN \l__xeCJK_aligni_tl \c__xeCJK_left_tl }
1982 }
1983 { \__xeCJK_if_last_punct_auxiv:TF }
1984 { \use_ii:nn }
1985 }
1986 \cs_new_protected:Npn \__xeCJK_if_last_punct_auxiv:TF
1987 {
1988   \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
1989   \xeCJK_if_last_node:TF
1990   {
1991     \tl_gset:Ne \g__xeCJK_last_punct_tl
1992     { \tex_Uchar:D \l__xeCJK_tmp_dim }
1993     \dim_set_eq:NN \l__xeCJK_last_bound_dim \l__xeCJK_last_kern_dim
1994     \use_i:nn
1995   }
1996   { \__xeCJK_make_node:N \l__xeCJK_tmp_dim \use_ii:nn }
1997 }
1998 \tl_new:N \l__xeCJK_aligni_tl
1999 \tl_new:N \l__xeCJK_alignii_tl
2000 \int_new:N \l__xeCJK_last_penalty_int
2001 \dim_new:N \l__xeCJK_last_bound_dim
2002 \bool_new:N \l__xeCJK_last_penalty_bool

\xeCJK_if_last_node:TF 2003 \cs_new_protected:Npn \xeCJK_if_last_node:TF #1#2
2004 {
2005   \__xeCJK_if_last_kern:TF
2006   {
2007     \dim_set_eq:NN \l__xeCJK_last_kern_dim \tex_lastkern:D
2008     \tex_unkern:D
2009     \__xeCJK_if_last_kern:TF
2010     {
2011       \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_last_kern_dim }
2012       { \tex_unkern:D #1 }
2013       { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
2014     }
2015     { \tex_kern:D \l__xeCJK_last_kern_dim #2 }
2016   }
2017   {#2}
2018 }
2019 \dim_new:N \l__xeCJK_last_kern_dim

\_xeCJK_CJK_and_FullRight_glue:N 2020 \cs_new_protected:Npn \__xeCJK_CJK_and_FullRight_glue:N #1
\_xeCJK_Default_and_FullRight_glue:N 2021 {
2022   \__xeCJK_punct_if_long:NTF #1
2023   {
2024     \__xeCJK_punct_if_nobreak_long:NTF #1
2025     { \xeCJK_no_break: }
2026     { \xeCJK_allow_break: }
2027   }
2028   { \xeCJK_no_break: }
2029   \__xeCJK_punct_if_middle:NT #1
2030   {
2031     \CJKglue
2032     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
2033     \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
2034   }
2035 }
2036 \cs_new_protected:Npn \__xeCJK_Default_and_FullRight_glue:N #1
2037 {
2038   \__xeCJK_punct_if_long:NTF #1
2039   {
2040     \__xeCJK_punct_if_nobreak_long:NTF #1
2041     { \xeCJK_no_break: }

```

```

2042         { \xeCJK_allow_break: }
2043     }
2044     { \xeCJK_no_break: }
2045     \_xeCJK_punct_if_middle:NT #1
2046     {
2047         \_xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
2048         \_xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
2049     }
2050 }

\xeCJK_FullLeft_and_FullLeft:N 2051 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullLeft:N #1
2052 {
2053     \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
2054     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
2055     \_xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
2056     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
2057     \xeCJK_fallback_punct_symbol:NN
2058     \CJKpunctsymbol #1
2059 }

\xeCJK_FullLeft_and_FullRight:N 2060 \cs_new_protected:Npn \xeCJK_FullLeft_and_FullRight:N #1
2061 {
2062     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
2063     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
2064     \_xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
2065     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
2066     \xeCJK_FullRight_symbol:N #1
2067 }

\xeCJK_FullRight_and_FullLeft:N 2068 \cs_new_protected:Npn \xeCJK_FullRight_and_FullLeft:N #1
2069 {
2070     \xeCJK_get_punct_bounds:NN \c__xeCJK_left_tl #1
2071     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
2072     \xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
2073     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
2074     \xeCJK_fallback_punct_symbol:NN
2075     \CJKpunctsymbol #1
2076 }

\xeCJK_FullRight_and_FullRight:N 2077 \cs_new_protected:Npn \xeCJK_FullRight_and_FullRight:N #1
2078 {
2079     \xeCJK_get_punct_bounds:NN \c__xeCJK_right_tl #1
2080     \xeCJK_get_punct_kerning:oN \g__xeCJK_last_punct_tl #1
2081     \_xeCJK_punct_kern:NN \g__xeCJK_last_punct_tl #1
2082     \tl_gset:Nn \g__xeCJK_last_punct_tl {#1}
2083     \xeCJK_FullRight_symbol:N #1
2084 }

```

5.7 全角右标点后的断行

CheckFullRight 选项设置。

```

2085 \keys_define:nn { xeCJK / options }
2086 {
2087     CheckFullRight .choice: ,
2088     CheckFullRight / true .code:n =
2089     {
2090         \cs_if_eq:NNF \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
2091         {
2092             \cs_set_eq:NN \_xeCJK_save_FullRight_check:
2093             \xeCJK_FullRight_and_Boundary:
2094             \cs_set_eq:NN \_xeCJK_save_FullRight_symbol:N
2095             \xeCJK_FullRight_symbol:N
2096             \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
2097             \xeCJK_check_FullRight:
2098             \cs_set_eq:NN \xeCJK_FullRight_symbol:N
2099             \xeCJK_check_FullRight_symbol:Nw
2100         }

```

```

2101     } ,
2102     CheckFullRight / false .code:n =
2103     {
2104         \cs_if_eq:NNT \xeCJK_FullRight_and_Boundary: \xeCJK_check_FullRight:
2105         {
2106             \cs_set_eq:NN \xeCJK_FullRight_and_Boundary:
2107                 \__xeCJK_save_FullRight_check:
2108             \cs_set_eq:NN \xeCJK_FullRight_symbol:N
2109                 \__xeCJK_save_FullRight_symbol:N
2110         }
2111     } ,
2112     CheckFullRight .default:n = { true }
2113 }

```

```

experiment/halfright-prebreakpenalty 2114 \keys_define:nn { xeCJK / options / experiment }
experiment/punct-measure-fix 2115 {
2116     halfright-prebreakpenalty .bool_set:N = \l__xeCJK_halfright_nobreak_bool ,
2117     halfright-prebreakpenalty .default:n = true ,
2118     halfright-prebreakpenalty .initial:n = false ,
2119     punct-measure-fix .bool_gset:N = \g__xeCJK_punct_measure_fix_bool ,
2120     punct-measure-fix .default:n = true ,
2121     punct-measure-fix .initial:n = false
2122 }

```

```

\xeCJK_FullRight_symbol:N 2123 \cs_new_protected:Npn \xeCJK_FullRight_symbol:N
2124 {
2125     \xeCJK_fallback_punct_symbol:NN
2126     \CJKpunctsymbol
2127 }

```

```

\xeCJK_check_FullRight: 2128 \cs_new_protected:Npn \xeCJK_check_FullRight:
2129 {
2130     \xeCJK_get_punct_bounds:No \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
2131     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
2132     \group_align_safe_begin:
2133     \token_case_meaning:NoTF \l_peek_token
2134     { \l__xeCJK_no_break_cs_case_tl }
2135     {
2136         \group_align_safe_end:
2137         \xeCJK_no_break:
2138         \group_insert_after:N \xeCJK_no_break:
2139     }
2140     { \group_align_safe_end: }
2141     \exp_after:wN \xeCJK_punct_node:N \g__xeCJK_last_punct_tl
2142     \xeCJK_class_group_end:
2143     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
2144     \__xeCJK_punct_boundary_guard:
2145 }
2146 \prg_generate_conditional_variant:Nnn \token_case_meaning:Nn { No } { TF , F }

```

```

\xeCJK_check_FullRight_symbol:Nw 2147 \cs_new_protected:Npn \xeCJK_check_FullRight_symbol:Nw #1
2148 { \peek_remove_spaces:n { \__xeCJK_save_FullRight_symbol:N #1 } }

```

```

\xeCJK_cs_case_keys_define:nNNnn 2149 \cs_new_protected:Npn \xeCJK_cs_case_keys_define:nNNnn #1#2#3#4#5
2150 {
2151     \tl_new:N #2
2152     \seq_new:N #3
2153     \keys_define:nn { xeCJK / options }
2154     {
2155         #1 .code:n =
2156         {
2157             \seq_set_split:Nnn #3 { } {##1}
2158             \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
2159         } ,
2160         #1+ .code:n =
2161         {
2162             \tl_map_inline:nn {##1}
2163             { \seq_if_in:NnF #3 {####1} { \seq_put_right:Nn #3 {####1} } }
2164             \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}

```

```

2165     } ,
2166     #1- .code:n =
2167     {
2168         \tl_map_inline:nn {##1} { \seq_remove_all:Nn #3 {###1} }
2169         \__xeCJK_update_cs_case_tl:NNnn #2#3 {#4} {#5}
2170     }
2171 }
2172 }
2173 \cs_new_protected:Npn \__xeCJK_update_cs_case_tl:NNnn #1#2#3#4
2174 {
2175     \tl_clear:N #1
2176     \seq_map_inline:Nn #2 { \tl_put_right:Nn #1 { {##1} {#3} } }
2177     #4
2178 }

```

NoBreakCS 设置不能在全角右标点之后断行的控制序列。

```

2179 \xeCJK_cs_case_keys_define:nnnn { NoBreakCS }
2180 \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。

```

2181 \NewDocumentCommand \xeCJKnobreak { }
2182 {
2183     \bool_set_true:N \l__xeCJK_tmp_bool
2184     \int_while_do:nnnn \tex_lastnodetype:D = \c__xeCJK_glue_node
2185     {
2186         \bool_if:NTF \l__xeCJK_tmp_bool
2187         {
2188             \bool_set_false:N \l__xeCJK_tmp_bool
2189             \skip_set_eq:NN \l__xeCJK_last_skip \tex_lastskip:D
2190         }
2191         { \skip_add:Nn \l__xeCJK_last_skip \tex_lastskip:D }
2192         \tex_unskip:D
2193     }
2194     \xeCJK_if_last_node:TF
2195     {
2196         \dim_set_eq:NN \l__xeCJK_tmp_dim \l__xeCJK_last_kern_dim
2197         \xeCJK_if_last_node:TF
2198         {
2199             \__xeCJK_if_last_glue:T
2200             {
2201                 \exp_args:NNNo \tex_unskip:D \xeCJK_no_break:
2202                 \skip_horizontal:n { \skip_use:N \tex_lastskip:D }
2203             }
2204             \__xeCJK_make_node:N \l__xeCJK_last_kern_dim
2205         }
2206         { }
2207         \__xeCJK_make_node:N \l__xeCJK_tmp_dim
2208     }
2209     { }
2210     \xeCJK_no_break:
2211     \bool_if:NF \l__xeCJK_tmp_bool
2212     { \skip_horizontal:N \l__xeCJK_last_skip }
2213 }

```

5.8 段末孤字处理

CheckSingle 孤字处理功能选项。

```

2214 \keys_define:nn { xeCJK / options }
2215 {
2216     CheckSingle .choice: ,
2217     CheckSingle / true .code:n =
2218     {
2219         \cs_if_eq:NNF \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
2220         {

```

```

2221         \cs_set_eq:NN \__xeCJK_check_single_save:N \xeCJK_CJK_and_CJK:N
2222         \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
2223     }
2224 } ,
2225 CheckSingle / false .code:n =
2226 {
2227     \cs_if_eq:NNT \xeCJK_CJK_and_CJK:N \xeCJK_check_single:Nw
2228     { \cs_set_eq:NN \xeCJK_CJK_and_CJK:N \__xeCJK_check_single_save:N }
2229 } ,
2230 CheckSingle .default:n = { true } ,
2231 CJKchecksingle .meta:n = { CheckSingle = true }
2232 }

```

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

```

2233 \keys_define:nn { xeCJK / options }
2234 {
2235     WidowPenalty .int_set:N = \l__xeCJK_widow_penalty_int ,
2236     WidowPenalty .default:n = { 10 000 }
2237 }

```

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

```

2238 \cs_new_protected:Npn \xeCJK_widow_penalty:
2239 { \tex_penalty:D \l__xeCJK_widow_penalty_int }

```

```

\__xeCJK_check_single:Nw 2240 \cs_new_protected:Npn \xeCJK_check_single:Nw #1
\__xeCJK_check_single_end:N 2241 {
2242     \group_align_safe_begin:
2243     \peek_catcode:NTF \c_catcode_letter_token
2244     { \xeCJK_check_single:NNw #1 }
2245     {
2246         \token_if_other:NTF \l_peek_token
2247         { \xeCJK_check_single:NNw }
2248         { \__xeCJK_check_single_end:N }
2249     } #1
2250 }
2251 }
2252 \cs_new_protected:Npn \__xeCJK_check_single_end:N
2253 {
2254     \group_align_safe_end:
2255     \__xeCJK_check_single_save:N
2256 }

```

`\xeCJK_check_single:NNw` 使用 `\group_align_safe_begin:` 和 `\group_align_safe_end:` 是为了防止在表格里面报错。
`\__xeCJK_check_single_aux:nnw`

```

2257 \cs_new_protected:Npn \xeCJK_check_single:NNw #1#2
2258 {
2259     \xeCJK_peek_catcode_ignore_spaces:NTF \c_catcode_letter_token
2260     {
2261         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
2262         {
2263             \bool_if:NTF \l__xeCJK_reserve_space_bool
2264             { \__xeCJK_check_single_end:N #1 #2 ~ }
2265             { \__xeCJK_check_single_space:NN #1#2 }
2266         }
2267         { \__xeCJK_check_single_end:N #1 #2 }
2268     }
2269     {
2270         \token_if_other:NTF \l_peek_token
2271         {
2272             \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
2273             { \__xeCJK_check_single_space:NN }
2274             { \__xeCJK_check_single_end:N }
2275         }
2276     }

```

```

2277         \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
2278         { \__xeCJK_check_single_aux:nNnw { ~ } }
2279         { \__xeCJK_check_single_aux:nNnw { } }
2280     }
2281     #1 #2
2282 }
2283 }
2284 \cs_new_protected:Npn \__xeCJK_check_single_aux:nNnw #1#2#3
2285 {
2286     \token_if_cs:NTF \l_peek_token
2287     { \xeCJK_check_single_cs:NNn }
2288     { \xeCJK_check_single_end:NNnw }
2289     #2 #3 {#1}
2290 }

```

```

\xeCJK_check_single_end:NNnw 2291 \cs_new_protected:Npn \__xeCJK_check_single_end_aux:NNn #1#2#3
\__xeCJK_check_single_end_aux:NNn 2292 { \__xeCJK_check_single_end:N #1 #2 #3 }
\__xeCJK_check_single_end_equation:NNnw 2293 \cs_new_eq:NN \xeCJK_check_single_end:NNnw \__xeCJK_check_single_end_aux:NNn
2294 \cs_new_protected:Npn \__xeCJK_check_single_end_equation:NNnw
2295 {
2296     \token_if_math_toggle:NTF \l_peek_token
2297     { \xeCJK_check_single_equation:NNnw }
2298     { \__xeCJK_check_single_end_aux:NNn }
2299 }

```

```

PlainEquation 2300 \keys_define:nn { xeCJK / options }
2301 {
2302     PlainEquation .choice: ,
2303     PlainEquation / true .code:n =
2304     {
2305         \cs_set_eq:NN \xeCJK_check_single_end:NNnw
2306         \__xeCJK_check_single_end_equation:NNnw
2307     } ,
2308     PlainEquation / false .code:n =
2309     {
2310         \cs_set_eq:NN \xeCJK_check_single_end:NNnw
2311         \__xeCJK_check_single_end_aux:NNn
2312     } ,
2313     PlainEquation .default:n = { true } ,
2314 }

```

```

\__xeCJK_check_single_space:NN 2315 \cs_new_protected:Npn \__xeCJK_check_single_space:NN #1#2
2316 {
2317     \xeCJK_if_CJK_class:NTF #2
2318     {
2319         \xeCJK_if_CJK_class:NTF \l_peek_token
2320         { \__xeCJK_check_single_end:N #1 #2 }
2321         { \__xeCJK_check_single_end:N #1 #2 ~ }
2322     }
2323     { \__xeCJK_check_single_end:N #1 #2 ~ }
2324 }

```

```

\xeCJK_check_single_equation:NNnw 2325 \cs_new_protected:Npn \xeCJK_check_single_equation:NNnw #1#2#3#4
2326 {
2327     \peek_catcode:NTF \c_math_toggle_token
2328     {
2329         \xeCJK_widow_penalty: \__xeCJK_check_single_end:N #1
2330         \xeCJK_make_node:n { CJK-widow } #2 #4
2331     }
2332     { \__xeCJK_check_single_end:N #1 #2#3#4 }
2333 }

```

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

```

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

```

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

```

2334 \cs_new_protected:Npn \xeCJK_check_single_cs:NNn #1#2#3
2335 {
2336   \token_case_meaning:Nof \l_peek_token
2337   { \l__xeCJK_check_single_cs_case_tl }
2338   { \use_iii:nnn }
2339   { \xeCJK_check_single_env:nnNn }
2340   {
2341     \xeCJK_widow_penalty:
2342     \__xeCJK_check_single_end:N #1
2343     \xeCJK_make_node:n { CJK-widow } #2#3
2344   }
2345   { \__xeCJK_check_single_end:N #1 #2#3 }
2346 }
2347 \tl_new:N \l__xeCJK_check_single_cs_case_tl
\xeCJK_check_single_env:nnNn 2348 \cs_new_protected:Npn \xeCJK_check_single_env:nnNn #1#2#3#4
2349 {
2350   \str_case_e:noTF {#4}
2351   { \l__xeCJK_inline_env_case_tl }
2352   {#2}
2353   {#1}
2354   #3 {#4}
2355 }
2356 \prg_generate_conditional_variant:Nnn \str_case_e:nn { no } { TF }

```

```

NewLineCS 2357 \xeCJK_cs_case_keys_define:nNNnn { NewLineCS }
2358 \l__xeCJK_new_line_cs_case_tl \l__xeCJK_new_line_cs_seq
2359 { \use_ii:nnn }
2360 {
2361   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
2362   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
2363 }

```

```

EnvCS 2364 \xeCJK_cs_case_keys_define:nNNnn { EnvCS }
2365 \l__xeCJK_env_cs_case_tl \l__xeCJK_env_cs_seq
2366 { \use:n }
2367 {
2368   \tl_concat:NNN \l__xeCJK_check_single_cs_case_tl
2369   \l__xeCJK_new_line_cs_case_tl \l__xeCJK_env_cs_case_tl
2370 }

```

```

InlineEnv 2371 \keys_define:nn { xeCJK / options }
2372 {
2373   InlineEnv .code:n =
2374   {
2375     \seq_set_from_clist:Nn \l__xeCJK_inline_env_seq {#1}
2376     \__xeCJK_update_inline_env_case_tl:
2377   },
2378   InlineEnv+ .code:n =
2379   {
2380     \clist_map_inline:nn {#1}
2381     {
2382       \seq_if_in:NnF \l__xeCJK_inline_env_seq {##1}
2383       { \seq_put_right:Nn \l__xeCJK_inline_env_seq {##1} }
2384     }
2385     \__xeCJK_update_inline_env_case_tl:
2386   },
2387   InlineEnv- .code:n =
2388   {
2389     \clist_map_inline:nn {#1}
2390     { \seq_remove_all:Nn \l__xeCJK_inline_env_seq {##1} }
2391     \__xeCJK_update_inline_env_case_tl:

```

```

2392     }
2393   }
2394   \seq_new:N \l__xeCJK_inline_env_seq
\__xeCJK_update_inline_env_case_tl: 2395   \cs_new_protected:Npn \__xeCJK_update_inline_env_case_tl:
2396   {
2397     \tl_clear:N \l__xeCJK_inline_env_case_tl
2398     \seq_map_inline:Nn \l__xeCJK_inline_env_seq
2399     { \tl_put_right:Nn \l__xeCJK_inline_env_case_tl { {##1} { } } }
2400   }
2401   \tl_new:N \l__xeCJK_inline_env_case_tl

```

5.9 增加 CJK 子分区

```

\g__xeCJK_CJK_sub_class_seq 2402 \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。
2403 \NewDocumentCommand \XeCJKDeclareSubCJKBlock
2404   { s > { \TrimSpaces } m m }
2405   {
2406     \XeCJK_declare_sub_char_class:nen { CJK } {#2} {#3}
2407     \IfBooleanT {#1} { \XeCJKResetPunctClass }
2408   }
2409   \@onlypreamble \XeCJKDeclareSubCJKBlock

\XeJKCancelSubCJKBlock 取消和恢复对 CJK 子区的声明。
\XeCJKRestoreSubCJKBlock
2410 \bool_new:N \l__xeCJK_sub_cancel_bool
2411 \NewDocumentCommand \XeJKCancelSubCJKBlock { s m }
2412   {
2413     \bool_if:NF \l__xeCJK_sub_cancel_bool
2414     {
2415       \bool_set_true:N \l__xeCJK_sub_cancel_bool
2416       \__xeCJK_sub_restore_or_cancel:e {#2}
2417       \IfBooleanT {#1} { \XeCJKResetPunctClass }
2418     }
2419   }
2420 \NewDocumentCommand \XeCJKRestoreSubCJKBlock { s m }
2421   {
2422     \bool_if:NT \l__xeCJK_sub_cancel_bool
2423     {
2424       \bool_set_false:N \l__xeCJK_sub_cancel_bool
2425       \__xeCJK_sub_restore_or_cancel:e {#2}
2426       \IfBooleanT {#1} { \XeCJKResetPunctClass }
2427     }
2428   }

\__xeCJK_sub_restore_or_cancel:n 2429 \cs_new_protected:Npn \__xeCJK_sub_restore_or_cancel:n #1
2430   {
2431     \clist_map_inline:nn {#1}
2432     {
2433       \int_if_exist:cTF { \__xeCJK_class_csname:n { CJK/##1 } }
2434       {
2435         \XeCJK_declare_char_class:nn
2436         { CJK \bool_if:NF \l__xeCJK_sub_cancel_bool { /##1 } }
2437         { \use:c { g__xeCJK_CJK/##1_range_clist } }
2438       }
2439       { \__xeCJK_error:ne { SubBlock-undefined } {##1} }
2440     }
2441   }
2442 \cs_generate_variant:Nn \__xeCJK_sub_restore_or_cancel:n { e }
2443 \__xeCJK_msg_new:nn { SubBlock-undefined }
2444   {
2445     The~CJK~sub~block~`#1'~is~undefined.\\\
2446     Try~to~use~\token_to_str:N \XeCJKDeclareSubCJKBlock \
2447     to~declare~it.
2448   }

```

```

\XeCJK_declare_sub_char_class:nnn 2449 \cs_new_protected:Npn \XeCJK_declare_sub_char_class:nnn #1#2#3
2450 {
2451   \int_if_exist:cF { \__XeCJK_class_csname:n { #1/#2 } }
2452   {
2453     \XeCJK_new_class:n { #1/#2 }
2454     \__XeCJK_set_sub_class_toks:nn {#1} {#2}
2455     \XeCJK_new_sub_key:n {#2}
2456   }
2457   \XeCJK_declare_char_class:nn { #1/#2 } {#3}
2458 }
2459 \cs_generate_variant:Nn \XeCJK_declare_sub_char_class:nnn { ne }

\__XeCJK_set_sub_class_toks:nn 2460 \cs_new_protected:Npn \__XeCJK_set_sub_class_toks:nn #1#2
2461 {
2462   \seq_map_inline:Nn \g__XeCJK_base_class_seq
2463   {
2464     \XeCJK_copy_inter_class_toks:nnnn { #1/#2 } {##1} {#1} {##1}
2465     \XeCJK_copy_inter_class_toks:nnnn {##1} { #1/#2 } {##1} {#1}
2466     \str_if_eq:nnTF {##1} { CJK }
2467     {
2468       \XeCJK_pre_inter_class_toks:nnn {##1} { #1/#2 }
2469       { \__XeCJK_switch_font:nn {#1} {#2} }
2470     }
2471     {
2472       \XeCJK_replace_inter_class_toks:nnnn {##1} { #1/#2 }
2473       { \XeCJK_fallback_symbol:NN }
2474       {
2475         \__XeCJK_switch_font:nn {#1} {#2}
2476         \XeCJK_fallback_symbol:NN
2477       }
2478     }
2479   }
2480   \XeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/#2 } {#1} {#1}
2481   \seq_map_inline:Nn \g__XeCJK_CJK_sub_class_seq
2482   {
2483     \XeCJK_copy_inter_class_toks:nnnn { #1/#2 } { #1/##1 } {#1} {#1}
2484     \XeCJK_copy_inter_class_toks:nnnn { #1/##1 } { #1/#2 } {#1} {#1}
2485     \XeCJK_pre_inter_class_toks:nnn { #1/#2 } { #1/##1 }
2486     { \__XeCJK_switch_font:nn {#2} {##1} }
2487     \XeCJK_pre_inter_class_toks:nnn { #1/##1 } { #1/#2 }
2488     { \__XeCJK_switch_font:nn {##1} {#2} }
2489   }
2490   \seq_gput_right:Nn \g__XeCJK_CJK_sub_class_seq {#2}
2491   \__XeCJK_save_CJK_class:n { #1/#2 }
2492   \clist_map_inline:nn
2493   {
2494     CJK , FullLeft , FullRight , HangulJamo ,
2495     HangulJamoL , HangulJamoV , HangulJamoT , CJStarter
2496   }
2497   {
2498     \XeCJK_pre_inter_class_toks:nnn { #1/#2 } {##1}
2499     { \__XeCJK_switch_font:nn {#2} {#1} }
2500   }
2501 }

```

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。

```

2502 \cs_if_exist:NF \tex_XeTeXglyphbounds:D
2503 {
2504   \__XeCJK_msg_new:nn { XeTeX-too-old }
2505   {
2506     \token_to_str:N \tex_XeTeXglyphbounds:D \ is~not~defined.\\
2507     CJK~punctuation~kerning~will~not~be~available.\\

```

```

2508     You~have~to~update~XeTeX~to~the~version~0.9995.0~or~later.
2509   }
2510   \__xeCJK_error:n { XeTeX-too-old }
2511   \AtEndOfPackage
2512   {
2513     \keys_define:nn { xeCJK / options }
2514     {
2515       PunctStyle .code:n =
2516       { \__xeCJK_error:ne { punct-style-unknown } {#1} }
2517     }
2518     \seq_gclear:N \g__xeCJK_punct_style_seq
2519     \__xeCJK_set_punct_style:n { plain }
2520   }
2521 }

```

\xeCJKsetwidth 手动设置参数中的标点符号的宽度。

```

2522 \NewDocumentCommand \xeCJKsetwidth { s m m }
2523 {
2524   \IfBooleanTF {#1}
2525   {
2526     \tl_map_inline:en {#2}
2527     {
2528       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
2529       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
2530       \tl_gset:cn { g__xeCJK_punct_bound_width/##1/tl } {#3}
2531     }
2532   }
2533   {
2534     \tl_map_inline:en {#2}
2535     {
2536       \tl_if_exist:cF { g__xeCJK_punct_width/##1/tl }
2537       { \tl_new:c { g__xeCJK_punct_width/##1/tl } }
2538       \tl_gset:cn { g__xeCJK_punct_width/##1/tl } {#3}
2539     }
2540   }
2541 }
2542 \@onlypreamble \xeCJKsetwidth
2543 \cs_generate_variant:Nn \tl_map_inline:nn { e }

```

\xeCJKsetkern 手动设置相邻标点的距离。

```

2544 \NewDocumentCommand \xeCJKsetkern { m m m }
2545 { \tl_gset:cn { g__xeCJK_punct/kern/#1/#2/tl } {#3} }
2546 \@onlypreamble \xeCJKsetkern

```

```

\c__xeCJK_left_tl 2547 \tl_const:Nn \c__xeCJK_left_tl { left }
\c__xeCJK_right_tl 2548 \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
2549 \keys_define:nn { xeCJK / options }
2550 {
2551   AllowBreakBetweenPuncts .choice: ,
2552   AllowBreakBetweenPuncts / true .code:n =
2553   {
2554     \bool_set_true:N \l__xeCJK_punct_breakable_bool
2555     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_breakable_kern:NN
2556     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
2557     \__xeCJK_punct_bound_breakable_kern:N
2558   } ,
2559   AllowBreakBetweenPuncts / false .code:n =
2560   {
2561     \bool_set_false:N \l__xeCJK_punct_breakable_bool
2562     \cs_set_eq:NN \xeCJK_punct_kern:NN \__xeCJK_punct_kern:NN
2563     \cs_set_eq:NN \__xeCJK_punct_bound_kern:N
2564     \__xeCJK_nobreak_hskip:N
2565   } ,

```

```

2566 AllowBreakBetweenPuncts .default:n = { true } ,
2567 KaiMingPunct .code:n = { \__xeCJK_set_special_punct:nn { mixed_width } {#1} } ,
2568 KaiMingPunct+ .code:n = { \__xeCJK_add_special_punct:nn { mixed_width } {#1} } ,
2569 KaiMingPunct- .code:n = { \__xeCJK_sub_special_punct:nn { mixed_width } {#1} } ,
2570 LongPunct .code:n = { \__xeCJK_set_special_punct:nn { long } {#1} } ,
2571 LongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { long } {#1} } ,
2572 LongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { long } {#1} } ,
2573 NoBreakLongPunct .code:n = { \__xeCJK_set_special_punct:nn { nobreak_long } {#1} } ,
2574 NoBreakLongPunct+ .code:n = { \__xeCJK_add_special_punct:nn { nobreak_long } {#1} } ,
2575 NoBreakLongPunct- .code:n = { \__xeCJK_sub_special_punct:nn { nobreak_long } {#1} } ,
2576 MiddlePunct .code:n = { \__xeCJK_set_special_punct:nn { middle } {#1} } ,
2577 MiddlePunct+ .code:n = { \__xeCJK_add_special_punct:nn { middle } {#1} } ,
2578 MiddlePunct- .code:n = { \__xeCJK_sub_special_punct:nn { middle } {#1} } ,
2579 PoZheHaoLigature .choice: ,
2580 PoZheHaoLigature / true .code:n =
2581 {
2582   \bool_set_true:N \l__xeCJK_pozhehao_ligature_bool
2583   \xeCJK_declare_char_class:nn { PoZheHao } { "2014" , "2015" }
2584 } ,
2585 PoZheHaoLigature / false .code:n =
2586 {
2587   \bool_set_false:N \l__xeCJK_pozhehao_ligature_bool
2588   \xeCJK_declare_char_class:nn { FullRight } { "2014" }
2589   \xeCJK_declare_char_class:nn { Default } { "2015" }
2590 } ,
2591 PoZheHaoLigature .default:n = { true } ,
2592 LatinPunct .choice: ,
2593 LatinPunct / true .code:n =
2594 {
2595   \bool_set_true:N \l__xeCJK_latin_punct_bool
2596   \xeCJK_declare_char_class:nn { HalfLeft }
2597   { "2018" , "201C" }
2598   \xeCJK_declare_char_class:nn { HalfRight }
2599   { "00B7" , "2019" , "201D" , "2025" , "2026" , "2027" }
2600 } ,
2601 LatinPunct / false .code:n =
2602 {
2603   \bool_set_false:N \l__xeCJK_latin_punct_bool
2604   \xeCJK_declare_char_class:nn { FullLeft }
2605   { "2018" , "201C" }
2606   \xeCJK_declare_char_class:nn { FullRight }
2607   { "00B7" , "2019" , "201D" , "2025" , "2026" , "2027" }
2608 } ,
2609 LatinPunct .default:n = { true } ,
2610 CJLineBreak .choices:nn = { normal , strict }
2611 {
2612   \str_if_eq:nnTF {#1} { strict }
2613   {
2614     \bool_set_true:N \l__xeCJK_CJ_strict_bool
2615     \xeCJK_declare_char_class:nN { CJStarter } \c__xeCJK_CJ_chars_clist
2616   }
2617   {
2618     \bool_set_false:N \l__xeCJK_CJ_strict_bool
2619     \xeCJK_declare_char_class:nN { CJK } \c__xeCJK_CJ_chars_clist
2620   }
2621 } ,
2622 CJLineBreak .initial:n = { normal } ,
2623 PunctWidth .tl_gset:N = \g__xeCJK_punct_width_tl ,
2624 PunctBoundWidth .tl_gset:N = \g__xeCJK_punct_bound_width_tl ,
2625 PunctWidth .value_required:n = true ,
2626 PunctBoundWidth .value_required:n = true ,
2627 RubberPunctSkip .choice: ,
2628 RubberPunctSkip .default:n = { true } ,
2629 RubberPunctSkip / true .code:n =
2630 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN } ,
2631 RubberPunctSkip / plus .code:n =
2632 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_plus:nNN } ,

```

```

2633 RubberPunctSkip / minus .code:n =
2634 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip_minus:nNN } ,
2635 RubberPunctSkip / false .code:n =
2636 { \cs_set_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_dim:nNN }
2637 }
2638 \bool_new:N \l__xeCJK_punct_breakable_bool

```

相关选项定义的辅助函数。

```

2639 \clist_new:N \g__xeCJK_special_punct_clist
2640 \clist_gset:Nn \g__xeCJK_special_punct_clist { mixed_width , long , nobreak_long , middle }
2641 \cs_new:Npn \__xeCJK_special_punct_seq:n #1 { g__xeCJK_special_punct_#1_seq }
2642 \cs_new:Npn \__xeCJK_special_punct_tl:nN #1#2 { g__xeCJK_special_punct_#1_#2_tl }
2643 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
2644 { \seq_new:c { \__xeCJK_special_punct_seq:n {#1} } }
2645 \cs_new_protected:Npn \__xeCJK_set_special_punct:nn #1#2
2646 {
2647   \seq_map_inline:cn { \__xeCJK_special_punct_seq:n {#1} }
2648   { \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} } }
2649   \seq_gclear:c { \__xeCJK_special_punct_seq:n {#1} }
2650   \tl_map_inline:en {#2}
2651   {
2652     \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
2653     \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
2654   }
2655 }
2656 \cs_new_protected:Npn \__xeCJK_add_special_punct:nn #1#2
2657 {
2658   \tl_map_inline:en {#2}
2659   {
2660     \seq_if_in:cnF { \__xeCJK_special_punct_seq:n {#1} } {##1}
2661     {
2662       \tl_new:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
2663       \seq_gput_right:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
2664     }
2665   }
2666 }
2667 \cs_new_protected:Npn \__xeCJK_sub_special_punct:nn #1#2
2668 {
2669   \tl_map_inline:en {#2}
2670   {
2671     \cs_undefine:c { \__xeCJK_special_punct_tl:nN {#1} {##1} }
2672     \seq_gremove_all:cn { \__xeCJK_special_punct_seq:n {#1} } {##1}
2673   }
2674 }

```

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。

```

2675 \prg_new_conditional:Npnn \__xeCJK_punct_if_right:N #1 { p , T , F , TF }
2676 {
2677   \if_int_compare:w \xeCJK_token_value_class:N #1 =
2678   \xeCJK_class_num:n { FullRight }
2679   \prg_return_true:
2680   \else:
2681     \if_int_compare:w \xeCJK_token_value_class:N #1 =
2682     \xeCJK_class_num:n { PoZheHao }
2683     \prg_return_true:
2684     \else:
2685       \prg_return_false:
2686     \fi:
2687   \fi:
2688 }
2689 \clist_map_inline:Nn \g__xeCJK_special_punct_clist
2690 {
2691   \exp_args:Nc
2692   \prg_new_conditional:Npnn { \__xeCJK_punct_if_#1:N } ##1 { p , T , F , TF }
2693   {

```

```

2694         \if_cs_exist:w \__xeCJK_special_punct_tl:nN {#1} {##1} \cs_end:
2695         \prg_return_true: \else: \prg_return_false: \fi:
2696     }
2697 }

```

一些用于记录的辅助函数。

```

2698 \cs_new:Npn \__xeCJK_punct_csname:n #1
2699 { c__xeCJK_l__xeCJK_current_punct_font_tl/l__xeCJK_punct_style_tl/#1/tl }
2700 \cs_new:Npn \__xeCJK_use_punct_dim:nN #1#2
2701 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2 } } }
2702 \cs_new:Npn \__xeCJK_use_punct_dim:nNN #1#2#3
2703 { \use:c { \__xeCJK_punct_csname:n { dim/#1/#2/#3 } } }
2704 \cs_new:Npn \__xeCJK_use_punct_skip:nNN #1#2#3
2705 { \use:c { \__xeCJK_punct_csname:n { skip/#1/#2/#3 } } }
2706 \cs_new:Npn \__xeCJK_use_punct_skip_plus:nNN #1#2#3
2707 { \use:c { \__xeCJK_punct_csname:n { skip/plus/#1/#2/#3 } } }
2708 \cs_new:Npn \__xeCJK_use_punct_skip_minus:nNN #1#2#3
2709 { \use:c { \__xeCJK_punct_csname:n { skip/minus/#1/#2/#3 } } }
2710 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNn #1#2
2711 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2 } }
2712 \cs_new_protected:Npn \__xeCJK_save_punct_dim:nNNn #1#2#3
2713 { \__xeCJK_save_punct_width_aux:nnnn { dim } {#1} { #1/#2/#3 } }
2714 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNn #1#2#3#4
2715 {
2716     \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { #1/#2/#3 } {#4}
2717     \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { plus/#1/#2/#3 } {#4}
2718     \__xeCJK_save_punct_width_aux:nnnn { skip } {#1} { minus/#1/#2/#3 } {#4}
2719 }
2720 \cs_new_protected:Npn \__xeCJK_save_punct_skip:nNNnnn #1#2#3#4#5#6
2721 {
2722     \exp_last_unbraced:Ne
2723     \__xeCJK_save_punct_skip_aux:nnnnn
2724     {
2725         {#1}
2726         { #1/#2/#3 }
2727         { \dim_eval:n {#4} }
2728         { \dim_max:nn { \c_zero_dim } {#5} }
2729         { \dim_max:nn { \c_zero_dim } {#6} }
2730     }
2731 }
2732 \cs_new_protected:Npn \__xeCJK_save_punct_skip_aux:nnnnn #1#2#3#4#5
2733 {
2734     \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
2735     {#2} { #3 ~ plus ~ #4 ~ minus ~ #5 ~ }
2736     \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
2737     { plus/#2 } { #3 ~ plus ~ #4 ~ }
2738     \__xeCJK_save_punct_width_aux:nnnn { skip } {#1}
2739     { minus/#2 } { #3 ~ minus ~ #5 ~ }
2740 }
2741 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:nnnn #1#2#3#4
2742 {
2743     \__xeCJK_save_punct_width_aux:cen
2744     { \__xeCJK_punct_csname:n { #1/#3 } }
2745     { \use:c { #1_eval:n } {#4} }
2746     {#2}
2747 }
2748 \cs_new_protected:Npn \__xeCJK_save_punct_width_aux:Nnn #1#2#3
2749 {
2750     \tl_const:Nn #1 {#2}
2751     \str_if_eq:nnT {#3} { glue }
2752     { \prop_gput:Nnn \g__xeCJK_punct_skip_prop {#2} { } }
2753 }
2754 \prop_new:N \g__xeCJK_punct_skip_prop
2755 \prop_gput:Non \g__xeCJK_punct_skip_prop { \skip_use:N \c_zero_skip } { }
2756 \cs_generate_variant:Nn \__xeCJK_save_punct_width_aux:Nnn { ce }
2757 \cs_new_eq:NN \__xeCJK_use_dim_or_skip:nNN \__xeCJK_use_punct_skip:nNN

```

定义标点处理模板。

```

2758 \DeclareObjectType { xeCJK / punctuation } { 0 }

2759 \DeclareTemplateInterface { xeCJK / punctuation } { basic } { 0 }
2760 {
2761     enabled-global-setting      : boolean = true ,
2762     fixed-punct-width          : length  = \c_max_dim ,
2763     fixed-punct-ratio          : real    = \c_one_fp ,
2764     mixed-punct-width          : length  = \KeyValue { fixed-punct-width } ,
2765     mixed-punct-ratio          : real    = \KeyValue { fixed-punct-ratio } ,
2766     middle-punct-width         : length  = \KeyValue { fixed-punct-width } ,
2767     middle-punct-ratio         : real    = \KeyValue { fixed-punct-ratio } ,
2768     fixed-margin-width         : length  = \c_max_dim ,
2769     fixed-margin-ratio         : real    = \c_one_fp ,
2770     mixed-margin-width         : length  = \KeyValue { fixed-margin-width } ,
2771     mixed-margin-ratio         : real    = \KeyValue { fixed-margin-ratio } ,
2772     middle-margin-width        : length  = \KeyValue { fixed-margin-width } ,
2773     middle-margin-ratio        : real    = \KeyValue { fixed-margin-ratio } ,
2774     bound-punct-width          : length  = \c_max_dim ,
2775     bound-punct-ratio          : real    = \c_nan_fp ,
2776     bound-margin-width         : length  = \c_max_dim ,
2777     bound-margin-ratio         : real    = \c_zero_fp ,
2778     enabled-hanging            : boolean = false ,
2779     add-min-bound-to-margin    : boolean = false ,
2780     optimize-margin            : boolean = false ,
2781     margin-minimum             : length  = \c_zero_dim ,
2782     enabled-kerning            : boolean = true ,
2783     enabled-left-right-kerning : boolean = true ,
2784     min-bound-to-kerning       : boolean = false ,
2785     kerning-total-width        : length  = \c_max_dim ,
2786     kerning-total-ratio        : real    = 0.75 ,
2787     optimize-kerning           : boolean = false ,
2788     same-align-margin          : length  = \c_max_dim ,
2789     same-align-ratio           : real    = \c_nan_fp ,
2790     different-align-margin     : length  = \c_max_dim ,
2791     different-align-ratio      : real    = \c_nan_fp ,
2792     kerning-margin-width       : length  = \c_max_dim ,
2793     kerning-margin-ratio       : real    = \c_one_fp ,
2794     kerning-margin-minimum     : length  = \c_zero_dim
2795 }

2796 \DeclareTemplateCode { xeCJK / punctuation } { basic } { 0 }
2797 {
2798     enabled-global-setting      = \l__xeCJK_enabled_global_setting_bool ,
2799     fixed-punct-width          = \l__xeCJK_fixed_punct_width_dim ,
2800     fixed-punct-ratio          = \l__xeCJK_fixed_punct_ratio_fp ,
2801     mixed-punct-width          = \l__xeCJK_mixed_punct_width_dim ,
2802     mixed-punct-ratio          = \l__xeCJK_mixed_punct_ratio_fp ,
2803     middle-punct-width         = \l__xeCJK_middle_punct_width_dim ,
2804     middle-punct-ratio         = \l__xeCJK_middle_punct_ratio_fp ,
2805     fixed-margin-width         = \l__xeCJK_fixed_margin_width_dim ,
2806     fixed-margin-ratio         = \l__xeCJK_fixed_margin_ratio_fp ,
2807     mixed-margin-width         = \l__xeCJK_mixed_margin_width_dim ,
2808     mixed-margin-ratio         = \l__xeCJK_mixed_margin_ratio_fp ,
2809     middle-margin-width        = \l__xeCJK_middle_margin_width_dim ,
2810     middle-margin-ratio        = \l__xeCJK_middle_margin_ratio_fp ,
2811     bound-punct-width          = \l__xeCJK_bound_punct_width_dim ,
2812     bound-punct-ratio          = \l__xeCJK_bound_punct_ratio_fp ,
2813     bound-margin-width         = \l__xeCJK_bound_margin_width_dim ,
2814     bound-margin-ratio         = \l__xeCJK_bound_margin_ratio_fp ,
2815     enabled-hanging            = \l__xeCJK_enabled_hanging_bool ,
2816     add-min-bound-to-margin    = \l__xeCJK_add_min_bound_to_margin_bool ,
2817     optimize-margin            = \l__xeCJK_optimize_margin_bool ,
2818     margin-minimum             = \l__xeCJK_margin_minimum_dim ,
2819     enabled-kerning            = \l__xeCJK_enabled_kerning_bool ,
2820     enabled-left-right-kerning = \l__xeCJK_enabled_left_right_kerning_bool ,
2821     min-bound-to-kerning       = \l__xeCJK_min_bound_to_kerning_bool ,
2822     kerning-total-width        = \l__xeCJK_kerning_total_width_dim ,

```

```

2823 kerning-total-ratio      = \l__xeCJK_kerning_total_ratio_fp ,
2824 optimize-kerning        = \l__xeCJK_optimize_kerning_bool ,
2825 same-align-margin       = \l__xeCJK_same_align_margin_dim ,
2826 same-align-ratio        = \l__xeCJK_same_align_ratio_fp ,
2827 different-align-margin   = \l__xeCJK_different_align_margin_dim ,
2828 different-align-ratio    = \l__xeCJK_different_align_ratio_fp ,
2829 kerning-margin-width     = \l__xeCJK_kerning_margin_width_dim ,
2830 kerning-margin-ratio     = \l__xeCJK_kerning_margin_ratio_fp ,
2831 kerning-margin-minimum   = \l__xeCJK_kerning_margin_minimum_dim
2832 }
2833 { \AssignTemplateKeys }

```

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

```

2834 \cs_new_protected:Npn \xeCJK_get_punct_bounds:NN #1#2
2835 {
2836   \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/glue/#1/#2 } }
2837   { \__xeCJK_get_punct_bounds_aux:NN #1 #2 }
2838 }
2839 \cs_new_protected:Npn \__xeCJK_get_punct_bounds_aux:NN
2840 {
2841   \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
2842   { \__xeCJK_save_punct_margin_plain:NN }
2843   { \__xeCJK_save_punct_margin:NN }
2844 }
2845 \cs_new_protected:Npn \xeCJK_get_punct_bounds:No
2846 { \exp_last_unbraced:NNo \xeCJK_get_punct_bounds:NN }
2847 \cs_new_protected:Npn \__xeCJK_save_punct_margin_plain:NN #1#2
2848 {
2849   \__xeCJK_save_punct_dim:nNNn { glue } #1 #2 { \c_zero_dim }
2850   \__xeCJK_save_punct_dim:nNNn { offset } #1 #2 { \c_zero_dim }
2851   \__xeCJK_save_punct_dim:nNNn { margin } #1 #2 { \c_zero_dim }
2852   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_left_tl {#2} { \c_zero_dim }
2853   \__xeCJK_save_punct_dim:nNNn { rule } \c__xeCJK_right_tl {#2} { \c_zero_dim }
2854   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_left_tl {#2} { \c_zero_dim }
2855   \__xeCJK_save_punct_dim:nNNn { bound } \c__xeCJK_right_tl {#2} { \c_zero_dim }
2856   \__xeCJK_save_punct_skip:nNNn { glue } #1 #2 { \c_zero_skip }
2857 }
2858 \cs_new_protected:Npn \__xeCJK_save_punct_margin:NN #1#2
2859 {
2860   \group_begin:
2861     \xeCJK_select_punct_font:
2862     \xeCJK_fallback_punct_symbol:NN
2863     \xeCJK_calc_punct_dimen:N #2
2864   \group_end:
2865   \dim_set:Nn \l__xeCJK_bound_dim
2866   { \__xeCJK_use_punct_dim:nNN { bound } #1 #2 }
2867   \tl_if_eq:NNTF #1 \c__xeCJK_right_tl
2868   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_left_tl }
2869   { \tl_set_eq:NN \l__xeCJK_reverse_tl \c__xeCJK_right_tl }
2870   \dim_set:Nn \l__xeCJK_reverse_bound_dim
2871   { \__xeCJK_use_punct_dim:nNN { bound } \l__xeCJK_reverse_tl #2 }
2872   \UseInstance { xeCJK / punctuation } { \l__xeCJK_punct_style_tl }
2873   \xeCJK_punct_margin_process:NN #1 #2
2874   \xeCJK_punct_offset_process:NN #1 #2
2875   \__xeCJK_punct_if_long:NT #2
2876   { \__xeCJK_long_punct_kerning:N #2 }
2877 }
2878 \tl_new:N \l__xeCJK_reverse_tl
2879 \dim_new:N \l__xeCJK_bound_dim
2880 \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。

```

2881 \cs_new_protected:Npn \__xeCJK_long_punct_kerning:N #1
2882 {
2883   \dim_set:Nn \l__xeCJK_tmp_dim
2884   {
2885     \dim_max:nn
2886     { \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim }
2887     { \c_zero_dim }
2888   }
2889   \__xeCJK_save_punct_dim:nNNn { bound_width } #1 #1 { \l__xeCJK_tmp_dim }
2890   \dim_set:Nn \l__xeCJK_tmp_dim
2891   {
2892     \str_case:nnF {#1}
2893     {
2894       { ~~~~2014 }
2895       {
2896         - \dim_max:nn
2897         { \l__xeCJK_tmp_dim }
2898         {
2899           \dim_max:nn
2900           {
2901             \__xeCJK_use_punct_dim:nN { dimen } #1 +
2902             \__xeCJK_use_punct_dim:nN { width } #1 -
2903             \f@size pt - \f@size pt
2904           }
2905           {
2906             \__xeCJK_use_punct_dim:nN { width } #1 +
2907             \__xeCJK_use_punct_dim:nN { width } #1 -
2908             \f@size pt - \f@size pt
2909           }
2910         }
2911       }
2912       { ~~~~2025 } { \c_zero_dim }
2913       { ~~~~2026 } { \c_zero_dim }
2914     }
2915     { -\l__xeCJK_tmp_dim }
2916   }
2917   \__xeCJK_save_punct_dim:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
2918   \__xeCJK_save_punct_skip:nNNn { kern } #1 #1 { \l__xeCJK_tmp_dim }
2919   \dim_add:Nn \l__xeCJK_tmp_dim
2920   { \dim_max:nn { \l__xeCJK_bound_dim } { \c_zero_dim } }
2921   \__xeCJK_save_punct_dim:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
2922   \__xeCJK_save_punct_skip:nNNn { bound_kern } #1 #1 { \l__xeCJK_tmp_dim }
2923 }

```

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。

```

2924 \cs_new_protected:Npn \xeCJK_get_punct_kerning:NN #1#2
2925 {
2926   \tl_if_exist:cF { \__xeCJK_punct_csname:n { dim/kern/#1/#2 } }
2927   {
2928     \tl_if_eq:NNTF \l__xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
2929     { \__xeCJK_save_punct_kerning_plain:NN }
2930     { \__xeCJK_save_punct_kerning:NN }
2931     #1 #2
2932   }
2933 }
2934 \cs_new_protected:Npn \xeCJK_get_punct_kerning:oN
2935 { \exp_after:wN \xeCJK_get_punct_kerning:NN }
2936 \cs_new_protected:Npn \__xeCJK_save_punct_kerning_plain:NN #1#2
2937 {

```

```

2938 \__xeCJK_save_punct_dim:nNn { kern } #1 #2 { \c_zero_dim }
2939 \__xeCJK_save_punct_dim:nNn { bound_kern } #1 #2 { \c_zero_dim }
2940 \__xeCJK_save_punct_dim:nNn { bound_width } #1 #2 { \c_zero_dim }
2941 \__xeCJK_save_punct_skip:nNn { kern } #1 #2 { \c_zero_skip }
2942 \__xeCJK_save_punct_skip:nNn { bound_kern } #1 #2 { \c_zero_skip }
2943 }
2944 \cs_new_protected:Npn \__xeCJK_save_punct_kerning:NN #1#2
2945 {
2946   % 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。
2947   \UseInstance { xeCJK / punctuation } { \l_xeCJK_punct_style_tl }
2948   \__xeCJK_punct_if_right:NTF #1
2949     { \xeCJK_punct_kerning_process:NN #1 #2 }
2950     {
2951       \__xeCJK_punct_if_right:NTF #2
2952       { \__xeCJK_save_left_right_kerning:NN #1 #2 }
2953       { \xeCJK_punct_kerning_process:NN #1 #2 }
2954     }
2955 }
2956 \cs_new_protected:Npn \__xeCJK_save_left_right_kerning:NN #1#2
2957 {
2958   \bool_if:NTF \l__xeCJK_enabled_left_right_kerning_bool
2959     { \xeCJK_punct_kerning_process:NN #1 #2 }
2960     {
2961       \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
2962       {
2963         \tl_if_exist:cTF { g__xeCJK_punct/kern/#1/#2/tl }
2964         { \xeCJK_punct_kerning_process:NN #1 #2 }
2965         { \__xeCJK_save_punct_kerning_plain:NN #1 #2 }
2966       }
2967       { \__xeCJK_save_punct_kerning_plain:NN #1 #2 }
2968     }
2969 }

```

__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。

```

2970 \prg_new_conditional:Npnn \__xeCJK_punct_if_full_margin_dash:N #1 { p , T , F , TF }
2971 {
2972   \str_if_eq:nnTF {#1} { ^^^2014 }
2973   {
2974     \if_int_compare:w \xeCJK_token_value_class:N #1 =
2975       \xeCJK_class_num:n { PoZheHao }
2976     \prg_return_false:
2977     \else:
2978     \prg_return_true:
2979     \fi:
2980   }
2981   { \prg_return_false: }
2982 }

```

\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。

```

2983 \cs_new_protected:Npn \xeCJK_punct_margin_process:NN #1#2
2984 {
2985   \dim_set:Nn \l__xeCJK_tmp_dim
2986   {
2987     \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
2988     {
2989       \cs_if_exist_use:cF { g__xeCJK_punct_width/#2/tl }
2990       {
2991         \tl_if_empty:NTF \g__xeCJK_punct_width_tl
2992         { \__xeCJK_calc_punct_width:N #2 }
2993         { \g__xeCJK_punct_width_tl }
2994       }
2995     }

```

```

2996         { \__xeCJK_calc_punct_width:N #2 }
2997     }
2998     \dim_set:Nn \l__xeCJK_margin_dim
2999     {
3000         \dim_max:nn
3001         { \l__xeCJK_margin_minimum_dim }
3002         {
3003             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
3004             {
3005                 \__xeCJK_punct_if_middle:NTF #2
3006                 {
3007                     ( \l__xeCJK_tmp_dim
3008                     - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
3009                     )
3010                     \__xeCJK_punct_if_full_margin_dash:NF #2 { / 2 }
3011                 }
3012                 {
3013                     \bool_if:NTF \l__xeCJK_optimize_margin_bool
3014                     {
3015                         \dim_max:nn
3016                         {
3017                             \dim_min:nn
3018                             { \l__xeCJK_bound_dim }
3019                             { \l__xeCJK_reverse_bound_dim }
3020                         }
3021                     }
3022                     { \use:n }
3023                     {
3024                         \l__xeCJK_tmp_dim
3025                         - \l__xeCJK_reverse_bound_dim
3026                         - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
3027                     }
3028                 }
3029             }
3030             {
3031                 \bool_if:NTF \l__xeCJK_optimize_margin_bool
3032                 { \dim_min:nn { \l__xeCJK_bound_dim } }
3033                 { \use:n }
3034                 { \__xeCJK_calc_margin_width:N #2 }
3035             }
3036         }
3037     }
3038     \__xeCJK_save_punct_dim:nNn { margin } #1 #2 { \l__xeCJK_margin_dim }
3039 }
3040 \dim_new:N \l__xeCJK_margin_dim

\__xeCJK_calc_punct_width:N 3041 \cs_new:Npn \__xeCJK_calc_punct_width:N #1
3042 {
3043     \__xeCJK_punct_if_middle:NTF #1
3044     { \__xeCJK_punct_width_or_ratio:nN { middle } }
3045     {
3046         \__xeCJK_punct_if_mixed_width:NTF #1
3047         { \__xeCJK_punct_width_or_ratio:nN { mixed } }
3048         { \__xeCJK_punct_width_or_ratio:nN { fixed } }
3049     }
3050     #1
3051 }

\__xeCJK_calc_margin_width:N 3052 \cs_new:Npn \__xeCJK_calc_margin_width:N #1
3053 {
3054     \__xeCJK_punct_if_middle:NTF #1
3055     {
3056         \dim_compare:nNnTF \l__xeCJK_middle_margin_width_dim < \c_max_dim
3057         { \l__xeCJK_middle_margin_width_dim }
3058         {
3059             \__xeCJK_dim_ratio:Nn \l__xeCJK_middle_margin_ratio_fp
3060             { ( \l__xeCJK_bound_dim + \l__xeCJK_reverse_bound_dim ) / 2 }
3061         }

```

```

3062     }
3063     {
3064         \__xeCJK_punct_if_mixed_width:NTF #1
3065         { \__xeCJK_margin_width_or_ratio:n { mixed } }
3066         { \__xeCJK_margin_width_or_ratio:n { fixed } }
3067     }
3068 }

\__xeCJK_dim_ratio:Nn 3069 \cs_new:Npn \__xeCJK_dim_ratio:Nn #1#2
3070 { \fp_to_dim:n { #1 \dim_to_fp:n {#2} } }
3071 \cs_generate_variant:Nn \__xeCJK_dim_ratio:Nn { c }

\__xeCJK_punct_offset_process:NN 3072 \cs_new_protected:Npn \xeCJK_punct_offset_process:NN #1#2
3073 {
3074     \dim_set:Nn \l__xeCJK_tmp_dim
3075     {
3076         \bool_if:NTF \l__xeCJK_enabled_global_setting_bool
3077         {
3078             \cs_if_exist_use:cF { g__xeCJK_punct_bound_width/#2/tl }
3079             {
3080                 \tl_if_empty:NTF \g__xeCJK_punct_bound_width_tl
3081                 { \__xeCJK_punct_width_or_ratio:nN { bound } #2 }
3082                 { \g__xeCJK_punct_bound_width_tl }
3083             }
3084         }
3085         { \__xeCJK_punct_width_or_ratio:nN { bound } #2 }
3086     }
3087     \dim_set:Nn \l__xeCJK_tmp_dim
3088     {
3089         \bool_if:NTF \l__xeCJK_enabled_hanging_bool
3090         { \use:n }
3091         { \dim_max:nn { \l__xeCJK_margin_minimum_dim } }
3092         {
3093             \dim_compare:nNnTF \l__xeCJK_tmp_dim < \c_max_dim
3094             {
3095                 \__xeCJK_punct_if_middle:NTF #2
3096                 {
3097                     \l__xeCJK_tmp_dim
3098                     - \l__xeCJK_margin_dim
3099                     - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
3100                 }
3101                 {
3102                     \l__xeCJK_tmp_dim
3103                     - \l__xeCJK_reverse_bound_dim
3104                     - ( \__xeCJK_use_punct_dim:nN { dimen } #2 )
3105                 }
3106             }
3107         }
3108         \bool_if:NTF \l__xeCJK_optimize_margin_bool
3109         { \dim_min:nn { \l__xeCJK_bound_dim } }
3110         { \use:n }
3111         { \__xeCJK_margin_width_or_ratio:n { bound } }
3112     }
3113 }
3114 }
3115 \__xeCJK_save_punct_dim:nNn { offset } #1 #2
3116 { \l__xeCJK_tmp_dim }
3117 \__xeCJK_save_punct_dim:nNn { rule } #1 #2
3118 { \l__xeCJK_tmp_dim - \l__xeCJK_bound_dim }
3119 \__xeCJK_save_punct_dim:nNn { rule } \l__xeCJK_reverse_tl #2
3120 { \l__xeCJK_tmp_dim - \l__xeCJK_reverse_bound_dim }
3121 \__xeCJK_save_punct_dim:nNn { glue } #1 #2
3122 { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
3123 \__xeCJK_save_punct_skip:nNnn { glue } #1 #2
3124 { \l__xeCJK_margin_dim - \l__xeCJK_tmp_dim }
3125 {
3126     \__xeCJK_punct_if_middle:NTF #2

```

```

3127     {
3128         ( \__xeCJK_use_punct_dim:nN { width } #2 -
3129           \__xeCJK_use_punct_dim:nN { dimen } #2 )
3130         \__xeCJK_punct_if_full_margin_dash:Nf #2 { / 2 }
3131         - \l__xeCJK_margin_dim
3132     }
3133     { \l__xeCJK_bound_dim - \l__xeCJK_margin_dim }
3134 }
3135 {
3136     \__xeCJK_punct_if_middle:Nf #2
3137     { \l__xeCJK_margin_dim / 2 }
3138     { \l__xeCJK_margin_dim - \l__xeCJK_reverse_bound_dim }
3139 }
3140 }

```

```

\__xeCJK_punct_width_or_ratio:nN 3141 \cs_new:Npn \__xeCJK_punct_width_or_ratio:nN #1#2
3142 {
3143     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_punct_width_dim } } < \c_max_dim
3144     { \use:c { l__xeCJK_#1_punct_width_dim } }
3145     {
3146         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_punct_ratio_fp } }
3147         { \c_max_dim }
3148         {
3149             \__xeCJK_dim_ratio:cn
3150             { l__xeCJK_#1_punct_ratio_fp }
3151             { \__xeCJK_use_punct_dim:nN { width } #2 }
3152         }
3153     }
3154 }

```

```

\__xeCJK_margin_width_or_ratio:n 3155 \cs_new:Npn \__xeCJK_margin_width_or_ratio:n #1
3156 {
3157     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_margin_width_dim } } < \c_max_dim
3158     { \use:c { l__xeCJK_#1_margin_width_dim } }
3159     {
3160         \__xeCJK_dim_ratio:cn
3161         { l__xeCJK_#1_margin_ratio_fp }
3162         { \l__xeCJK_bound_dim }
3163     }
3164     \bool_if:NT \l__xeCJK_add_min_bound_to_margin_bool
3165     { + \dim_min:nn \l__xeCJK_bound_dim \l__xeCJK_reverse_bound_dim }
3166 }

```

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

```

3167 \cs_new_protected:Npn \xeCJK_punct_kerning_process:NN #1#2
3168 {
3169     \dim_set:Nn \l__xeCJK_margin_dim
3170     { \__xeCJK_original_kerning_margin:NN #1 #2 }
3171     \dim_set:Nn \l__xeCJK_minimum_bound_dim
3172     { \__xeCJK_punct_min_bound:NN #1 #2 }
3173     \__xeCJK_punct_if_long:Nf #1
3174     { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
3175     {
3176         \__xeCJK_punct_if_long:Nf #2
3177         { \bool_set_false:N \l__xeCJK_enabled_kerning_bool }
3178     }
3179     \dim_set:Nn \l__xeCJK_kerning_margin_dim
3180     {
3181         \bool_if:Nf \l__xeCJK_enabled_global_setting_bool
3182         {
3183             \cs_if_exist_use:cF { g__xeCJK_punct/kern/#1/#2/tl }
3184             { \__xeCJK_punct_kerning_process_aux:NN #1 #2 }
3185         }
3186         { \__xeCJK_punct_kerning_process_aux:NN #1 #2 }
3187     }
3188     \__xeCJK_save_kerning:nnNN { kern } { bound } #1 #2

```

```

3189 \__xeCJK_save_punct_dim:nNNn { bound_width } #1 #2
3190 { \l__xeCJK_kerning_margin_dim - \l__xeCJK_tmp_dim }
3191 \__xeCJK_punct_if_right:NTF #1
3192 {
3193   \__xeCJK_punct_if_right:NTF #2
3194   { \__xeCJK_save_kerning:nnnNN { bound_kern } { offset } { bound } }
3195   { \__xeCJK_save_kerning:nnNN { bound_kern } { offset } }
3196 }
3197 {
3198   \__xeCJK_punct_if_right:NTF #2
3199   { \__xeCJK_save_kerning:nnNN { bound_kern } { bound } }
3200   { \__xeCJK_save_kerning:nnnNN { bound_kern } { bound } { offset } }
3201 }
3202 #1 #2
3203 }
3204 \cs_new:Npn \__xeCJK_punct_kerning_process_aux:NN #1#2
3205 {
3206   \bool_if:NTF \l__xeCJK_enabled_kerning_bool
3207   { \__xeCJK_calc_kerning_margin:NN #1 #2 }
3208   { \l__xeCJK_margin_dim }
3209 }
3210 \dim_new:N \l__xeCJK_minimum_bound_dim
3211 \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。

```

3212 \cs_new_protected:Npn \__xeCJK_save_kerning:nnNN #1#2
3213 { \__xeCJK_save_kerning:nnnNN {#1} {#2} {#2} }
3214 \cs_new_protected:Npn \__xeCJK_save_kerning:nnnNN #1#2#3#4#5
3215 {
3216   \dim_set:Nn \l__xeCJK_tmp_dim
3217   {
3218     \l__xeCJK_kerning_margin_dim
3219     - ( \__xeCJK_use_punct_dim:nNN {#2} \c__xeCJK_right_tl #4 )
3220     - ( \__xeCJK_use_punct_dim:nNN {#3} \c__xeCJK_left_tl #5 )
3221   }
3222   \__xeCJK_save_punct_dim:nNNn {#1} #4 #5 { \l__xeCJK_tmp_dim }
3223   \__xeCJK_save_punct_skip:nNNnnn {#1} #4 #5
3224   { \l__xeCJK_tmp_dim }
3225   { \l__xeCJK_margin_dim - \l__xeCJK_kerning_margin_dim }
3226   { \l__xeCJK_kerning_margin_dim - \l__xeCJK_minimum_bound_dim }
3227 }

```

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

```

3228 \cs_new:Npn \__xeCJK_original_kerning_margin:NN #1#2
3229 {
3230   \dim_eval:n
3231   {
3232     \__xeCJK_use_punct_dim:nNN
3233     { \__xeCJK_punct_if_right:NTF #1 { margin } { bound } } \c__xeCJK_right_tl #1
3234     +
3235     \__xeCJK_use_punct_dim:nNN
3236     { \__xeCJK_punct_if_right:NTF #2 { bound } { margin } } \c__xeCJK_left_tl #2
3237   }
3238 }

```

```

\__xeCJK_calc_kerning_margin:NN 3239 \cs_new:Npn \__xeCJK_calc_kerning_margin:NN #1#2
\__xeCJK_calc_kerning_margin_aux:NN 3240 {
3241   \dim_max:nn
3242   { \l__xeCJK_kerning_margin_minimum_dim }
3243   {
3244     \bool_if:NTF \l__xeCJK_min_bound_to_kerning_bool
3245     { \l__xeCJK_minimum_bound_dim }
3246     {
3247       \bool_if:NTF \l__xeCJK_optimize_kerning_bool
3248       { \dim_max:nn { \l__xeCJK_minimum_bound_dim } }
3249       { \use:n }

```

```

3250         { \_xeCJK_calc_kerning_margin_aux:NN #1 #2 }
3251     }
3252 }
3253 }
3254 \cs_new:Npn \_xeCJK_calc_kerning_margin_aux:NN #1#2
3255 {
3256     \dim_compare:nNnTF \l__xeCJK_kerning_total_width_dim < \c_max_dim
3257     { \_xeCJK_calc_kerning_margin:nNN \l__xeCJK_kerning_total_width_dim }
3258     {
3259         \fp_if_nan:nTF { \l__xeCJK_kerning_total_ratio_fp }
3260         {
3261             \xeCJK_if_same_class:NNTF #1 #2
3262             { \_xeCJK_kerning_width_or_ratio:nNN { same } }
3263             { \_xeCJK_kerning_width_or_ratio:nNN { different } }
3264         }
3265         {
3266             \_xeCJK_calc_kerning_margin:nNN
3267             {
3268                 \_xeCJK_dim_ratio:Nn \l__xeCJK_kerning_total_ratio_fp
3269                 {
3270                     \_xeCJK_use_punct_dim:nN { width } #1 +
3271                     \_xeCJK_use_punct_dim:nN { width } #2
3272                 }
3273             }
3274         }
3275     }
3276     #1 #2
3277 }

```

```

\_xeCJK_kerning_width_or_ratio:nNN 3278 \cs_new:Npn \_xeCJK_kerning_width_or_ratio:nNN #1#2#3
3279 {
3280     \dim_compare:nNnTF { \use:c { l__xeCJK_#1_align_margin_dim } } < \c_max_dim
3281     { \use:c { l__xeCJK_#1_align_margin_dim } }
3282     {
3283         \fp_if_nan:nTF { \use:c { l__xeCJK_#1_align_ratio_fp } }
3284         {
3285             \dim_compare:nNnTF \l__xeCJK_kerning_margin_width_dim < \c_max_dim
3286             { \l__xeCJK_kerning_margin_width_dim \use_none:n }
3287             { \_xeCJK_dim_ratio:Nn \l__xeCJK_kerning_margin_ratio_fp }
3288         }
3289         { \_xeCJK_dim_ratio:cn { l__xeCJK_#1_align_ratio_fp } }
3290         { \l__xeCJK_margin_dim }
3291     }
3292 }

```

```

\_xeCJK_punct_min_bound:NN 3293 \cs_new:Npn \_xeCJK_punct_min_bound:NN #1#2
3294 {
3295     \dim_max:nn
3296     {
3297         \dim_min:nn
3298         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #1 }
3299         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #1 }
3300     }
3301     {
3302         \dim_min:nn
3303         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_left_tl #2 }
3304         { \_xeCJK_use_punct_dim:nNN { bound } \c__xeCJK_right_tl #2 }
3305     }
3306 }

```

_xeCJK_calc_kerning_margin:nNN #2 和 #3 为相邻的两个标点, #1 为要确定的相邻两个标点总共占的宽度。

```

3307 \cs_new:Npn \_xeCJK_calc_kerning_margin:nNN #1#2#3
3308 {
3309     \dim_eval:n
3310     {
3311         (#1)
3312         - ( \_xeCJK_use_punct_dim:nNN

```

```

3313         { \_xeCJK_punct_if_right:NTF #2 { bound } { margin } }
3314         \c_xeCJK_left_tl #2 )
3315     - ( \_xeCJK_use_punct_dim:nNn
3316         { \_xeCJK_punct_if_right:NTF #3 { margin } { bound } }
3317         \c_xeCJK_right_tl #3 )
3318     - ( \_xeCJK_use_punct_dim:nN { dimen } #2 )
3319     - ( \_xeCJK_use_punct_dim:nN { dimen } #3 )
3320 }
3321 }

```

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

```

3322 \cs_new_protected:Npn \xeCJK_calc_punct_dimen:N #1
3323 {
3324     \_xeCJK_save_punct_dim:nNn { bound } \c_xeCJK_left_tl #1
3325     { \xeCJK_glyph_bounds:NN 1 #1 }
3326     \_xeCJK_save_punct_dim:nNn { bound } \c_xeCJK_right_tl #1
3327     { \xeCJK_glyph_bounds:NN 3 #1 }
3328     \_xeCJK_save_punct_dim:nNn { width } #1
3329     { \tex_fontcharwd:D \tex_font:D `#1 }
3330     \_xeCJK_save_punct_dim:nNn { dimen } #1
3331     {
3332         ( \_xeCJK_use_punct_dim:nN { width } #1 ) -
3333         ( \_xeCJK_use_punct_dim:nNn { bound } \c_xeCJK_left_tl #1 ) -
3334         ( \_xeCJK_use_punct_dim:nNn { bound } \c_xeCJK_right_tl #1 )
3335     }
3336 }

```

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

```

3337 \cs_new:Npn \xeCJK_glyph_bounds:NN #1#2
3338 { \tex_XeTeXglyphbounds:D #1 ~ \tex_XeTeXcharglyph:D `#2 \exp_stop_f: }

```

```

PunctStyle 3339 \keys_define:nn { xeCJK / options }
3340 { PunctStyle .code:n = \exp_args:Ne \_xeCJK_set_punct_style:n {#1} }
3341 \cs_new_protected:Npn \_xeCJK_set_punct_style:n #1
3342 {
3343     \IfInstanceExistTF { xeCJK / punctuation } {#1}
3344     { \tl_set:Nn \l_xeCJK_punct_style_tl {#1} }
3345     {
3346         \prop_get:NnNF \c_xeCJK_punct_style_alias_prop
3347         {#1} \l_xeCJK_punct_style_tl
3348         { \_xeCJK_error:ne { punct-style-unknown } {#1} }
3349     }
3350 }
3351 \prop_const_from_keyval:Nn \c_xeCJK_punct_style_alias_prop
3352 {
3353     halfwidth      = banjiao ,
3354     fullwidth      = quanjiao ,
3355     mixedwidth     = kaiming ,
3356     marginkerning  = hangmobanjiang ,
3357     plain          = plain
3358 }
3359 \tl_new:N \l_xeCJK_punct_style_tl
3360 \tl_const:Nn \c_xeCJK_punct_style_plain_tl { plain }
3361 \_xeCJK_msg_new:nn { punct-style-unknown }
3362 {
3363     Punctuation~style~"#1"~is~unknown. \\\
3364     The~available~styles~are~listed~as~follow.\\\
3365     "plain,~\seq_use:Nnnn \g_xeCJK_punct_style_seq
3366     { ~and~ } { ,~ } { ,~and~ }".\\
3367 }

```

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

```

3368 \cs_new_protected:Npn \_xeCJK_trim_spaces:n #1
3369 {
3370     \tl_set:Ne \ProcessedArgument

```

```

3371     { \exp_args:Ne \tl_trim_spaces:n {#1} }
3372   }

```

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

```

3373 \NewDocumentCommand \xeCJKDeclarePunctStyle
3374 { > { \__xeCJK_trim_spaces:n } m m }
3375 {
3376   \IfInstanceExistTF { xeCJK / punctuation } {#1}
3377   { \__xeCJK_warning:ne { punct-style-already-defined } {#1} }
3378   { \seq_gput_right:Nn \g__xeCJK_punct_style_seq {#1} }
3379   \DeclareInstance { xeCJK / punctuation } {#1} { basic } {#2}
3380 }
3381 \seq_new:N \g__xeCJK_punct_style_seq
3382 \__xeCJK_msg_new:nn { punct-style-already-defined }
3383 {
3384   Punctuation~style~"#1"~is~already~defined!. \\\
3385   The~existing~style~of~"#1"~will~be~overwritten.\\
3386 }
3387 \@onlypreamble \xeCJKDeclarePunctStyle

```

\xeCJKEditPunctStyle 对已有的标点处理风格进行修改。

```

3388 \NewDocumentCommand \xeCJKEditPunctStyle
3389 { > { \__xeCJK_trim_spaces:n } m m }
3390 {
3391   \IfInstanceExistTF { xeCJK / punctuation } {#1}
3392   { \EditInstance { xeCJK / punctuation } {#1} {#2} }
3393   { \__xeCJK_error:ne { punct-style-unknown } {#1} }
3394 }
3395 \@onlypreamble \xeCJKEditPunctStyle

```

默认设置即为全角格式。

```

3396 \xeCJKDeclarePunctStyle { quanjiao }
3397 {
3398   optimize-kerning          = true ,
3399   enabled-left-right-kerning = false
3400 }
3401 \xeCJKDeclarePunctStyle { hangmoban_jiao } { enabled-kerning = false }
3402 \xeCJKDeclarePunctStyle { ban_jiao }
3403 {
3404   fixed-punct-ratio    = 0.5 ,
3405   optimize-margin      = true ,
3406   kerning-total-ratio  = 0.5 ,
3407   optimize-kerning     = true
3408 }
3409 \xeCJKDeclarePunctStyle { kaiming }
3410 {
3411   fixed-punct-ratio    = 0.5 ,
3412   mixed-punct-ratio    = 1.0 ,
3413   optimize-margin      = true ,
3414   kerning-total-ratio  = 0.5 ,
3415   optimize-kerning     = true
3416 }
3417 \xeCJKDeclarePunctStyle { CCT }
3418 {
3419   fixed-punct-ratio    = 0.7 ,
3420   optimize-margin      = true ,
3421   kerning-total-ratio  = 0.6 ,
3422   optimize-kerning     = true
3423 }

```

5.11 后备字体

AutoFallBack 后备字体的宏包选项声明。

```

3424 \keys_define:nn { xeCJK / options }
3425 {
3426   AutoFallBack .choice: ,
3427   AutoFallBack / true .code:n =
3428   {
3429     \cs_set_eq:NN \xeCJK_fallback_symbol:NN
3430     \__xeCJK_fallback_symbol:NN
3431     \cs_set_eq:NN \xeCJK_fallback_punct_symbol:NN
3432     \__xeCJK_fallback_punct_symbol:NN
3433     \cs_set_eq:NN \xeCJK_clear_fallback_font:
3434     \__xeCJK_clear_fallback_font:
3435   } ,
3436   AutoFallBack / false .code:n =
3437   {
3438     \xeCJK_cs_clear:N \xeCJK_fallback_symbol:NN
3439     \xeCJK_cs_clear:N \xeCJK_fallback_punct_symbol:NN
3440     \xeCJK_cs_clear:N \xeCJK_clear_fallback_font:
3441   } ,
3442   AutoFallBack .default:n = { true } ,
3443   fallback .meta:n = { AutoFallBack = true }
3444 }

```

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
3445 \cs_new_protected:Npn \__xeCJK_fallback_symbol:NN #1#2
3446 {
3447   \xeCJK_reset_fallback_font:
3448   \xeCJK_glyph_if_exist:NF #2
3449   { \__xeCJK_fallback_symbol_aux:NN }
3450   #1#2
3451 }
3452 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol:NN #1#2
3453 {
3454   \xeCJK_reset_fallback_font:
3455   \xeCJK_glyph_if_exist:NF #2
3456   { \__xeCJK_fallback_punct_symbol_aux:NN }
3457   #1#2
3458 }
3459 \cs_new_eq:NN \xeCJK_fallback_symbol:NN \prg_do_nothing:
3460 \cs_new_eq:NN \xeCJK_fallback_punct_symbol:NN \prg_do_nothing:
3461 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:NN
3462 {
3463   \__xeCJK_fallback_symbol_aux:nnNN
3464   { \CJK@family }
3465   { \l_xeCJK_family_tl }
3466 }
3467 \cs_new_protected:Npn \__xeCJK_fallback_punct_symbol_aux:NN
3468 {
3469   \__xeCJK_fallback_symbol_aux:nnNN
3470   { \CJK@punctfamily }
3471   { \l_xeCJK_punct_family_tl }
3472 }
3473 \cs_new_protected:Npn \__xeCJK_fallback_symbol_aux:nnNN
3474 {
3475   \cs_set_protected:Npe \xeCJK_reset_fallback_font:
3476   {
3477     \tex_the:D \tex_font:D
3478     \xeCJK_clear_fallback_font:
3479   }
3480   \exp_args:Nee \__xeCJK_fallback_loop:nnNN
3481 }
3482 \cs_new_protected:Npn \__xeCJK_clear_fallback_font:
3483 { \cs_set_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing: }
3484 \cs_new_eq:NN \xeCJK_reset_fallback_font: \prg_do_nothing:

```

```
3485 \cs_new_eq:NN \xeCJK_clear_fallback_font: \prg_do_nothing:
```

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

```
3486 \cs_new_protected:Npn \_xeCJK_fallback_loop:nnNN
3487 {
3488   \cs_set_eq:NN \_xeCJK_fallback_loop:TF \use_i:nn
3489   \_xeCJK_fallback_loop:nnnNN { FallBack }
3490 }
3491 \cs_new_protected:Npn \_xeCJK_fallback_loop:nnnNN #1#2#3
3492 {
3493   \xeCJK_select_fallback_font:nnn {#1} {#2} {#3}
3494   \_xeCJK_fallback_loop:TF
3495     { \_xeCJK_fallback_loop_aux:nnnNN }
3496     { \_xeCJK_fallback_missing_glyph:nnnNN }
3497     {#1} {#2} {#3}
3498 }
3499 \cs_new_protected:Npn \_xeCJK_fallback_loop_aux:nnnNN #1#2#3#4#5
3500 {
3501   \xeCJK_glyph_if_exist:NF #5
3502   { \_xeCJK_fallback_loop:nnnNN { #1/FallBack } {#2} {#3} }
3503   #4#5
3504 }
3505 \cs_new_protected:Npn \_xeCJK_fallback_missing_glyph:nnnNN #1#2#3#4#5
3506 {
3507   \_xeCJK_warning:neee { missing-glyph } {#1} {#2} {#5}
3508   #4#5
3509 }
3510 \cs_new_protected:Npn \xeCJK_select_fallback_font:nnn #1#2
3511 {
3512   \_xeCJK_select_fallback_font:cnnn
3513   { \_xeCJK_font_csname:n { #2/#1 } } {#1} {#2}
3514 }
3515 \cs_new_protected:Npn \_xeCJK_select_fallback_font:Nnnn #1
3516 {
3517   \cs_if_exist:NF #1
3518   { \_xeCJK_fallback_font_initial:Nnnnn }
3519   #1 \use_none:nnn
3520 }
3521 \cs_generate_variant:Nn \_xeCJK_select_fallback_font:Nnnn { c }
3522 \cs_new_protected:Npn \_xeCJK_fallback_font_initial:Nnnnn #1#2#3#4#5
3523 {
3524   \xeCJK_family_if_exist:nTF { #5/#3 }
3525     { \_xeCJK_font_initial:Nn #1 { #5/#3 } }
3526     { \_xeCJK_fallback_font_initial_auxi:Nnnn #1 {#5} {#3} {#4} }
3527   #1
3528 }
3529 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxi:Nnnn #1
3530 {
3531   \exp_args:NNe \_xeCJK_fallback_font_initial_auxii:Nnnnn
3532   #1 { \CJKfamilydefault }
3533 }
3534 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxii:Nnnnn #1#2#3
3535 {
3536   \str_if_eq:nnTF {#2} {#3}
3537     { \_xeCJK_fallback_loop_end:Nnnn }
3538     { \_xeCJK_fallback_font_initial_auxiii:Nnnn }
3539     #1 {#2}
3540 }
3541 \cs_new_protected:Npn \_xeCJK_fallback_font_initial_auxiii:Nnnn #1#2
3542 {
3543   \xeCJK_family_if_exist:nTF {#2}
3544     { \_xeCJK_fallback_font_initial_auxiv:Nnnn }
3545     { \_xeCJK_fallback_loop_end:Nnnn }
3546     #1 {#2}
3547 }
```

```

3548 \cs_new_protected:Npn \__xeCJK_fallback_font_initial_auxiv:Nnnn #1#2#3#4
3549 {
3550   \__xeCJK_font_initial:Nn #1 {#2}
3551   \exp_args:Nc \__xeCJK_fallback_font_initial_auxiii:Nnnn
3552     { \__xeCJK_font_csname:n { #4/#3/FallBack } }
3553     { #2/FallBack } { #3/FallBack } {#4}
3554 }
3555 \cs_new_eq:NN \__xeCJK_fallback_loop:TF \use_i:nn
3556 \cs_new_protected:Npn \__xeCJK_fallback_loop_end:Nnnn #1#2#3#4
3557 { \cs_gset_eq:NN #1 \__xeCJK_fallback_loop_end: }
3558 \cs_new_protected:Npn \__xeCJK_fallback_loop_end:
3559 { \cs_set_eq:NN \__xeCJK_fallback_loop:TF \use_ii:nn }
3560 \__xeCJK_msg_new:nn { missing-glyph }
3561 {
3562   CJKfamily~\__xeCJK_msg_family_map:n {#2}'~(#{1})~
3563   does~not~contain~glyph~#{3}'~
3564   ( U + \int_to_Hex:n { `#3 } )~\msg_line_context:.
3565 }

```

```

\setCJKfallbackfamilyfont 3566 \NewDocumentCommand \setCJKfallbackfamilyfont { m o m }
3567 {
3568   \__xeCJK_pass_args:nnnn
3569   { \xeCJK_set_family_fallback:nnn {#1} } {#2} {#3}
3570   { }
3571 }

\xCJK_set_family_fallback:nnn 3572 \cs_new_protected:Npn \xeCJK_set_family_fallback:nnn #1#2#3
3573 {
3574   \group_begin:
3575   \tl_set:Nn \l__xeCJK_fallback_family_tl {#1}
3576   \prop_get:Nof \g__xeCJK_family_font_name_prop
3577     \l__xeCJK_fallback_family_tl \l__xeCJK_font_name_tl
3578     { \tl_clear:N \l__xeCJK_font_name_tl }
3579   \clist_set:Nn \l__xeCJK_public_options_clist {#2}
3580   \clist_map_function:nN {#3} \__xeCJK_set_family_fallback:n
3581   \group_end:
3582 }
3583 \cs_new_protected:Npn \__xeCJK_set_family_fallback:n #1
3584 {
3585   \tl_put_right:Nn \l__xeCJK_fallback_family_tl { /FallBack }
3586   \__xeCJK_get_sub_features:on \l__xeCJK_fallback_family_tl {#1}
3587   \clist_concat:NNN \l__xeCJK_sub_font_options_clist
3588     \l__xeCJK_public_options_clist
3589     \l__xeCJK_sub_font_options_clist
3590   \exp_args:Nooo \xeCJK_set_family:nnn
3591     \l__xeCJK_fallback_family_tl
3592     \l__xeCJK_sub_font_options_clist
3593     \l__xeCJK_sub_font_name_tl
3594 }
3595 \tl_new:N \l__xeCJK_fallback_family_tl
3596 \clist_new:N \l__xeCJK_public_options_clist

```

5.12 CJK 字体族声明方式

```

3597 \bool_new:N \g__xeCJK_auto_fake_bold_bool
3598 \bool_new:N \g__xeCJK_auto_fake_slant_bool
3599 \fp_new:N \g__xeCJK_embolden_factor_fp
3600 \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

```

3601 \keys_define:nn { xeCJK / options }
3602 {
3603   AutoFakeBold .choices:nn = { true , false }
3604   { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_bold_bool } ,
3605   AutoFakeBold / unknown .code:n =
3606   {
3607     \bool_gset_true:N \g__xeCJK_auto_fake_bold_bool

```

```

3608     \fp_gset:Nn \g__xeCJK_embolden_factor_fp { \l_keys_value_tl }
3609   } ,
3610   AutoFakeBold .default:n = { true } ,
3611   AutoFakeSlant .choices:nn = { true , false }
3612   { \use:c { bool_gset_ #1 :N } \g__xeCJK_auto_fake_slant_bool } ,
3613   AutoFakeSlant / unknown .code:n =
3614   {
3615     \bool_gset_true:N \g__xeCJK_auto_fake_slant_bool
3616     \fp_gset:Nn \g__xeCJK_slant_factor_fp { \l_keys_value_tl }
3617   } ,
3618   AutoFakeSlant .default:n = { true } ,
3619   EmboldenFactor .fp_gset:N = \g__xeCJK_embolden_factor_fp ,
3620   SlantFactor .fp_gset:N = \g__xeCJK_slant_factor_fp ,
3621   BoldFont .meta:n = { AutoFakeBold = true } ,
3622   boldfont .meta:n = { AutoFakeBold = true } ,
3623   SlantFont .meta:n = { AutoFakeSlant = true } ,
3624   slantfont .meta:n = { AutoFakeSlant = true }
3625 }

```

\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
3626 \seq_new:N \g__xeCJK_sub_key_seq
3627 \cs_new_protected:Npn \xeCJK_new_sub_key:n #1
3628 {
3629   \seq_gput_right:Nn \g__xeCJK_sub_key_seq {#1}
3630   \keys_define:nn { xeCJK / features }
3631   {
3632     #1 .code:n =
3633     {
3634       \tl_if_blank:nTF {##1}
3635       {
3636         \prop_clear:N \l__xeCJK_sub_key_prop
3637         \tl_set:N \l__xeCJK_sub_family_name_tl
3638         { \l__xeCJK_family_name_tl /#1 }
3639         \clist_remove_all:Nn \l__xeCJK_font_options_clist {#1}
3640       }
3641       {
3642         \tl_clear:N \l__xeCJK_sub_family_name_tl
3643         \str_if_eq:nnTF {##1} { * }
3644         { \prop_put:Nnn \l__xeCJK_sub_key_prop {#1} { \q_no_value } }
3645         { \__xeCJK_get_sub_features:nn {#1} {##1} }
3646       }
3647     } ,
3648     #1 .default:n = { }
3649   }
3650 }

```

```

\__xeCJK_get_sub_features:nn 3651 \cs_new_protected:Npn \__xeCJK_get_sub_features:nn #1#2
\__xeCJK_get_sub_features:w 3652 {
3653   \tl_set:N \l__xeCJK_tmp_tl { \xeCJK_tl_remove_outer_braces:n {#2} }
3654   \clist_clear:N \l__xeCJK_sub_font_options_clist
3655   \exp_after:wN \__xeCJK_get_sub_features:w \l__xeCJK_tmp_tl
3656   \q_mark [ \q_nil ] \q_mark \q_stop
3657   \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
3658   { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl }
3659   { \tl_replace_all:Nno \l__xeCJK_sub_font_name_tl { * } \l__xeCJK_font_name_tl }
3660   \prop_put:Nne \l__xeCJK_sub_key_prop {#1}
3661   {
3662     { \exp_not:o \l__xeCJK_sub_font_options_clist }
3663     { \exp_not:o \l__xeCJK_sub_font_name_tl }
3664   }
3665 }
3666 \cs_new_protected:Npn \__xeCJK_get_sub_features:w #1 [#2] #3 \q_mark #4 \q_stop
3667 {
3668   \quark_if_nil:nTF {#2}
3669   { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }

```

```

3670     {
3671         \tl_set:N \l__xeCJK_sub_font_name_tl
3672         { \xeCJK_tl_remove_outer_braces:n {#3} }
3673         \tl_if_empty:NTF \l__xeCJK_sub_font_name_tl
3674         { \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_tmp_tl }
3675         { \clist_set:Nn \l__xeCJK_sub_font_options_clist {#2} }
3676     }
3677 }
3678 \tl_new:N \l__xeCJK_sub_family_name_tl
3679 \tl_new:N \l__xeCJK_sub_font_name_tl
3680 \clist_new:N \l__xeCJK_sub_font_options_clist
3681 \cs_generate_variant:Nn \__xeCJK_get_sub_features:nn { o }
3682 \cs_generate_variant:Nn \tl_replace_all:Nnn { Nno }

Fallback 3683 \xeCJK_new_sub_key:n { Fallback }

BoldFont 调用字体的属性声明, 同 fontspec 宏包。
ItalicFont
3684 \keys_define:nn { xeCJK / features }
3685 {
3686     BoldFont .tl_set:N = \l__xeCJK_font_name_bf_tl ,
3687     ItalicFont .tl_set:N = \l__xeCJK_font_name_it_tl
3688 }

AutoFakeBold 3689 \keys_define:nn { xeCJK / features }
AutoFakeSlant 3690 {
3691     AutoFakeBold .choice: ,
3692     AutoFakeBold / true .code:n =
3693     {
3694         \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
3695         \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
3696     } ,
3697     AutoFakeBold / false .code:n =
3698     { \bool_set_false:N \l__xeCJK_auto_fake_bold_bool } ,
3699     AutoFakeBold / unknown .code:n =
3700     {
3701         \bool_set_true:N \l__xeCJK_auto_fake_bold_bool
3702         \fp_set:Nn \l__xeCJK_embolden_factor_fp { \l_keys_value_tl }
3703     } ,
3704     AutoFakeBold .default:n = { true } ,
3705     AutoFakeSlant .choice: ,
3706     AutoFakeSlant / true .code:n =
3707     {
3708         \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
3709         \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp
3710     } ,
3711     AutoFakeSlant / false .code:n =
3712     { \bool_set_false:N \l__xeCJK_auto_fake_slant_bool } ,
3713     AutoFakeSlant / unknown .code:n =
3714     {
3715         \bool_set_true:N \l__xeCJK_auto_fake_slant_bool
3716         \fp_set:Nn \l__xeCJK_slant_factor_fp { \l_keys_value_tl }
3717     } ,
3718     AutoFakeSlant .default:n = { true }
3719 }

__xeCJK_set_family_initial: 3720 \cs_new_protected:Npn __xeCJK_set_family_initial:
3721 {
3722     \int_gincr:N \g__xeCJK_family_int
3723     \prop_clear:N \l__xeCJK_sub_key_prop
3724     \tl_clear:N \l__xeCJK_font_name_bf_tl
3725     \tl_clear:N \l__xeCJK_font_name_it_tl
3726     \tl_clear:N \l__xeCJK_sub_family_name_tl
3727     \clist_clear:N \l__xeCJK_fontspec_options_clist
3728     \bool_set_eq:NN \l__xeCJK_auto_fake_bold_bool \g__xeCJK_auto_fake_bold_bool
3729     \bool_set_eq:NN \l__xeCJK_auto_fake_slant_bool \g__xeCJK_auto_fake_slant_bool
3730     \fp_set_eq:NN \l__xeCJK_embolden_factor_fp \g__xeCJK_embolden_factor_fp
3731     \fp_set_eq:NN \l__xeCJK_slant_factor_fp \g__xeCJK_slant_factor_fp

```

```

3732 }
3733 \int_new:N \g__xeCJK_family_int
3734 \prop_new:N \l__xeCJK_sub_key_prop
3735 \clist_new:N \l__xeCJK_fontspec_options_clist
3736 \bool_new:N \l__xeCJK_auto_fake_bold_bool
3737 \bool_new:N \l__xeCJK_auto_fake_slant_bool
3738 \fp_new:N \l__xeCJK_embolden_factor_fp
3739 \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。

```

3740 \cs_new_protected:Npn \xeCJK_set_family:nnn #1#2#3
3741 {
3742   \group_begin:
3743   \__xeCJK_set_family_initial:
3744   \tl_set:Nn \l__xeCJK_family_name_tl {#1}
3745   \clist_set:Nn \l__xeCJK_font_options_clist {#2}
3746   \tl_set:Nn \l__xeCJK_font_name_tl {#3}
3747   \clist_concat:NNN \l__xeCJK_font_options_clist
3748   \g__xeCJK_default_features_clist \l__xeCJK_font_options_clist
3749   \keys_set_known:noN { xeCJK / features }
3750   \l__xeCJK_font_options_clist \l__xeCJK_fontspec_options_clist
3751   \__xeCJK_binding_sub_family:
3752   \__xeCJK_parse_font_shape:
3753   \__xeCJK_check_family:o \l__xeCJK_family_name_tl
3754   \__xeCJK_gset_family_cs:n { \l__xeCJK_family_name_tl }
3755   \__xeCJK_save_family_info:
3756   \__xeCJK_set_sub_block_family:
3757   \group_end:
3758 }
3759 \tl_new:N \l__xeCJK_family_name_tl
3760 \tl_new:N \l__xeCJK_font_name_tl
3761 \clist_new:N \l__xeCJK_font_options_clist
3762 \cs_generate_variant:Nn \xeCJK_set_family:nnn { e , o }

```

```

\__xeCJK_binding_sub_family: 3763 \cs_new_protected:Npn \__xeCJK_binding_sub_family:
3764 {
3765   \tl_if_empty:NF \l__xeCJK_sub_family_name_tl
3766   { \tl_set_eq:NN \l__xeCJK_family_name_tl \l__xeCJK_sub_family_name_tl }
3767 }

```

```

\__xeCJK_gset_family_cs:n 3768 \cs_new_protected:Npn \__xeCJK_gset_family_cs:n #1
3769 {
3770   \cs_gset_protected:cpe { \__xeCJK_family_csname:n {#1} }
3771   {
3772     \group_begin:
3773     \exp_not:n { \cs_set_eq:NN \xeCJK@fontfamily \use_none:n }
3774     \exp_not:n { \fontspec_gset_family:Nnn \g__xeCJK_fontspec_family_tl }
3775     { \exp_not:o \l__xeCJK_fontspec_options_clist }
3776     { \exp_not:o \l__xeCJK_font_name_tl }
3777     \__xeCJK_gset_family_nfss_cs:no
3778     {#1} { \exp_not:N \g__xeCJK_fontspec_family_tl }
3779     \group_end:
3780     \tl_set_eq:NN \exp_not:N \l__xeCJK_fontspec_family_tl
3781     \exp_not:N \g__xeCJK_fontspec_family_tl
3782   }
3783 }
3784 \tl_new:N \g__xeCJK_fontspec_family_tl
3785 \tl_new:N \l__xeCJK_fontspec_family_tl

```

```

\__xeCJK_check_family:n 3786 \cs_new_protected:Npn \__xeCJK_check_family:n #1
3787 {
3788   \prop_gpop:NnNT \g__xeCJK_family_font_name_prop {#1} \l__xeCJK_tmp_tl
3789   {
3790     \prop_gpop:NnNT \g__xeCJK_family_name_prop {#1} \l__xeCJK_tmp_tl
3791     {
3792       \cs_undefine:c { \__xeCJK_family_csname:n {#1} }
3793       \cs_undefine:c { \__xeCJK_family_nfss_csname:n {#1} }

```

```

3794     }
3795     \__xeCJK_warning:nee { CJKfamily-redef } {#1} { \l__xeCJK_tmp_tl }
3796 }
3797 }
3798 \cs_generate_variant:Nn \__xeCJK_check_family:n { o }
3799 \__xeCJK_msg_new:nn { CJKfamily-redef }
3800 { Redefining~CJKfamily~\__xeCJK_msg_family_map:n {#1}'~(#2). }

\__xeCJK_parse_font_shape: 3801 \cs_new_protected:Npn \__xeCJK_parse_font_shape:
3802 {
3803     \tl_if_blank:oTF \l__xeCJK_font_name_bf_tl
3804     {
3805         \bool_if:NT \l__xeCJK_auto_fake_bold_bool
3806         {
3807             \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
3808             { AutoFakeBold = { \fp_use:N \l__xeCJK_embolden_factor_fp } }
3809         }
3810     }
3811     {
3812         \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
3813         { BoldFont = { \exp_not:o \l__xeCJK_font_name_bf_tl } }
3814     }
3815     \tl_if_blank:oTF \l__xeCJK_font_name_it_tl
3816     {
3817         \bool_if:NT \l__xeCJK_auto_fake_slant_bool
3818         {
3819             \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
3820             { AutoFakeSlant = { \fp_use:N \l__xeCJK_slant_factor_fp } }
3821         }
3822     }
3823     {
3824         \clist_put_right:Ne \l__xeCJK_fontspec_options_clist
3825         { ItalicFont = { \exp_not:o \l__xeCJK_font_name_it_tl } }
3826     }
3827 }

\g__xeCJK_family_name_prop 3828 \prop_new:N \g__xeCJK_family_name_prop
\g__xeCJK_family_font_name_prop 3829 \prop_new:N \g__xeCJK_family_font_name_prop
\g__xeCJK_family_font_options_prop 3830 \prop_new:N \g__xeCJK_family_font_options_prop

\__xeCJK_save_family_info: 3831 \cs_new_protected:Npn \__xeCJK_save_family_info:
3832 {
3833     \exp_args:Nooo \__xeCJK_save_family_info:nnn
3834     \l__xeCJK_family_name_tl
3835     \l__xeCJK_font_name_tl
3836     \l__xeCJK_font_options_clist
3837 }
3838 \cs_new_protected:Npn \__xeCJK_save_family_info:nnn #1#2#3
3839 {
3840     \prop_gput:Nnn \g__xeCJK_family_font_name_prop {#1} {#2}
3841     \prop_gput:Nnn \g__xeCJK_family_font_options_prop {#1} {#3}
3842 }

\__xeCJK_set_sub_block_family: 3843 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:
3844 {
3845     \prop_if_empty:NF \l__xeCJK_sub_key_prop
3846     {
3847         \prop_map_function:NN
3848         \l__xeCJK_sub_key_prop
3849         \__xeCJK_set_sub_block_family:nn
3850     }
3851 }
3852 \cs_new_protected:Npn \__xeCJK_set_sub_block_family:nn #1#2
3853 {
3854     \tl_set:Ne \l__xeCJK_sub_family_name_tl { \l__xeCJK_family_name_tl/#1 }
3855     \quark_if_no_value:nTF {#2}
3856     { \__xeCJK_copy_sub_family:n {#1} }
3857     { \xeCJK_set_family:onn \l__xeCJK_sub_family_name_tl #2 }

```

```

3858 }
3859 \cs_new_protected:Npn \__xeCJK_copy_sub_family:n #1
3860 {
3861   \__xeCJK_check_family:o \l__xeCJK_sub_family_name_tl
3862   \prop_get:NoNT \g__xeCJK_family_font_name_prop
3863     \l__xeCJK_family_name_tl \l__xeCJK_sub_font_name_tl
3864   {
3865     \prop_gput:Noo \g__xeCJK_family_font_name_prop
3866       \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_name_tl
3867   }
3868   \prop_get:NoNT \g__xeCJK_family_font_options_prop
3869     \l__xeCJK_family_name_tl \l__xeCJK_sub_font_options_clist
3870   {
3871     \clist_remove_all:Nn \l__xeCJK_sub_font_options_clist { #1 = * }
3872     \prop_gput:Noo \g__xeCJK_family_font_options_prop
3873       \l__xeCJK_sub_family_name_tl \l__xeCJK_sub_font_options_clist
3874   }
3875   \cs_gset_protected:cpe
3876     { \__xeCJK_family_csname:n { \l__xeCJK_sub_family_name_tl } }
3877   {
3878     \xeCJK_family_if_exist:eT { \l__xeCJK_family_name_tl }
3879     {
3880       \__xeCJK_gset_family_nfss_cs:no
3881         { \l__xeCJK_sub_family_name_tl }
3882         { \exp_not:N \l__xeCJK_fontspec_family_tl }
3883     }
3884   }
3885 }

```

```

\__xeCJK_copy_family:nn 3886 \cs_new_protected:Npn \__xeCJK_copy_family:nn #1#2
\__xeCJK_copy_family:ee 3887 {
3888   \xeCJK_family_if_exist:nT {#2}
3889   {
3890     \prop_gput:Nno \g__xeCJK_family_name_prop
3891       {#1} \l__xeCJK_fontspec_family_tl
3892     \tl_map_inline:nn
3893       {
3894         \g__xeCJK_family_font_name_prop
3895         \g__xeCJK_family_font_options_prop
3896       }
3897     {
3898       \prop_get:NnNT ##1 {#2} \l__xeCJK_tmp_tl
3899       { \prop_gput:Nno ##1 {#1} \l__xeCJK_tmp_tl }
3900     }
3901     \cs_gset_eq:cc
3902       { \__xeCJK_family_nfss_csname:n {#1} }
3903       { \__xeCJK_family_nfss_csname:n {#2} }
3904   }
3905 }
3906 \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
3907 \cs_new:Npn \__xeCJK_font_csname:n #1
3908 { xeCJK/#1/\f@series/\f@shape/\f@size }
3909 \tl_new:N \l__xeCJK_current_font_tl
3910 \tl_set:No \l__xeCJK_current_font_tl
3911 { \__xeCJK_font_csname:n { \CJK@family } }
3912 \cs_new_protected:Npn \xeCJK_select_font:
3913 {
3914   \__xeCJK_select_font:cn
3915     { \l__xeCJK_current_font_tl }
3916     { \l__xeCJK_family_tl }
3917 }
3918 \cs_new_protected:Npn \__xeCJK_select_font:Nn #1#2

```

```

3919 {
3920   \xeCJK_clear_fallback_font:
3921   \cs_if_exist:NF #1 { \__xeCJK_font_initial:Nn #1 {#2} }
3922   #1
3923 }
3924 \cs_generate_variant:Nn \__xeCJK_select_font:Nn { c }
3925 \tl_new:N \l__xeCJK_current_coor_tl
3926 \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)。

```

3927 \cs_new_protected:Npn \__xeCJK_font_initial:Nn #1#2
3928 {
3929   \group_begin:
3930   \__xeCJK_font_initial_hook:
3931   \__xeCJK_family_use:n {#2}
3932   \xeCJK_font_gset_to_current:N #1
3933   \group_end:
3934 }
3935 \cs_new_protected:Npn \__xeCJK_font_initial_hook:
3936 { \tl_use:N \g__xeCJK_font_initial_hook_tl }
3937 \cs_new_protected:Npn \__xeCJK_gadd_font_initial_hook:n
3938 { \tl_gput_right:Nn \g__xeCJK_font_initial_hook_tl }
3939 \tl_new:N \g__xeCJK_font_initial_hook_tl

```

\xeCJK_select_punct_font: 切换标点符号字体。

```

\l__xeCJK_current_punct_font_tl
3940 \cs_new_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
3941 \cs_new_protected:Npn \__xeCJK_select_punct_font_aux:
3942 {
3943   \__xeCJK_select_font:cn
3944   { \l__xeCJK_current_punct_font_tl }
3945   { \l__xeCJK_punct_family_tl }
3946 }
3947 \tl_new:N \CJK@punctfamily
3948 \tl_new:N \l__xeCJK_punct_family_tl
3949 \tl_new:N \l__xeCJK_current_punct_font_tl
3950 \tl_set:Nn \CJK@punctfamily { \CJK@family }
3951 \tl_set:Nn \l__xeCJK_punct_family_tl { \l__xeCJK_family_tl }
3952 \tl_set:Nn \l__xeCJK_current_punct_font_tl
3953 { \__xeCJK_font_csname:n { \CJK@punctfamily } }
3954 \cs_new_eq:NN \__xeCJK_select_font: \prg_do_nothing:
3955 \cs_new_eq:NN \__xeCJK_select_punct_font: \prg_do_nothing:

```

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

```

3956 \cs_new_protected:Npn \__xeCJK_switch_font:nn #1#2
3957 {
3958   \str_if_eq:nnF {#1} {#2}
3959   {
3960     \__xeCJK_info:nee { CJK-block } {#1} {#2}
3961     \str_if_eq:nnTF {#2} { CJK }
3962     { \xeCJK_select_font: }
3963     { \xeCJK_select_font:n {#2} }
3964   }
3965 }
3966 \__xeCJK_msg_new:nn { CJK-block } { Switch~from~block~`#1'~to~`#2'. }

```

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

```

3967 \cs_new_protected:Npn \xeCJK_select_font:n #1
3968 {
3969   \__xeCJK_select_font:cnn
3970   { \__xeCJK_font_csname:n { \CJK@family/#1 } }

```

```

3971     { \l_xeCJK_family_tl }
3972     {#1}
3973   }
3974   \cs_new_protected:Npn \__xeCJK_select_font:Nnn #1#2#3
3975   {
3976     \xeCJK_clear_fallback_font:
3977     \cs_if_exist:NF #1
3978     { \__xeCJK_block_font_initial:Nnn #1 {#2} {#3} }
3979     #1
3980   }
3981   \cs_generate_variant:Nn \__xeCJK_select_font:Nnn { c }
3982   \cs_new_protected:Npn \__xeCJK_block_font_initial:Nnn #1#2#3
3983   {
3984     \xeCJK_block_family:nn {#2} {#3}
3985     \__xeCJK_font_initial:Nn #1 { #2/#3 }
3986   }
3987   \cs_new_protected:Npn \xeCJK_block_family:nn #1#2
3988   {
3989     \xeCJK_family_if_exist:eF { #1/#2 }
3990     {
3991       \__xeCJK_copy_family:ee { #1/#2 }
3992       {
3993         \cs_if_exist:cTF
3994         { \__xeCJK_family_csname:n { \CJKfamilydefault/#2 } }
3995         { \CJKfamilydefault/#2 } {#1}
3996       }
3997     }
3998   }

\__xeCJK_family_csname:n 3999 \cs_new:Npn \__xeCJK_family_csname:n #1
\__xeCJK_family_nfss_csname:n 4000 { xeCJK/family/#1 }
\__xeCJK_family_use:n 4001 \cs_new:Npn \__xeCJK_family_nfss_csname:n #1
\__xeCJK_gset_family_nfss_cs:nn 4002 { xeCJK/family/nfss/#1 }
4003 \cs_new_protected:Npn \__xeCJK_family_use:n #1
4004 { \use:c { \__xeCJK_family_nfss_csname:n {#1} } }
4005 \cs_new_protected:Npn \__xeCJK_gset_family_nfss_cs:nn #1#2
4006 {
4007   \prop_gput:Nnn \g__xeCJK_family_name_prop {#1} {#2}
4008   \cs_gset_protected:cpe
4009   { \__xeCJK_family_nfss_csname:n {#1} }
4010   { \__xeCJK_nfss_family:nn { \c__xeCJK_encoding_tl } {#2} }
4011 }
4012 \cs_generate_variant:Nn \__xeCJK_gset_family_nfss_cs:nn { no }

```

__xeCJK_nfss_family:n 用于处理 L^AT_EX₂_ε 2020/02/02 中 \bfseries@rm 等与 \bfdefault 不一致可能导致的问题。

```

4013 \cs_if_exist:NTF \fontseriesforce
4014 {
4015   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
4016   {
4017     \fontencoding {#1}
4018     \str_if_eq:eeF { \f@series } { \bfdefault }
4019     {
4020       \str_case_e:nn { \f@family }
4021       {
4022         { \rmdefault } { \__xeCJK_nfss_series:n { rm } }
4023         { \sfdefault } { \__xeCJK_nfss_series:n { sf } }
4024         { \ttdefault } { \__xeCJK_nfss_series:n { tt } }
4025       }
4026     }
4027     \fontfamily {#2}
4028     \selectfont
4029   }
4030   \cs_new_protected:Npn \__xeCJK_nfss_series:n #1
4031   {
4032     \str_if_eq:eeT { \f@series } { \use:c { bfseries@#1 } }
4033     { \fontseriesforce { \bfdefault } }
4034   }

```

```

4035 }
4036 {
4037   \cs_new_protected:Npn \__xeCJK_nfss_family:nn #1#2
4038   {
4039     \fontencoding {#1}
4040     \tl_set:Nn \f@family {#2}
4041     \selectfont
4042   }
4043 }
\xeCJK_family_if_exist:nTF 4044 \prg_new_protected_conditional:Npnn \xeCJK_family_if_exist:n #1 { T , F , TF }
4045 {
4046   \prop_get:NnNTF \g__xeCJK_family_name_prop
4047   {#1} \l__xeCJK_fontspec_family_tl
4048   { \prg_return_true: }
4049   {
4050     \exp_args:Ne \cs_if_exist_use:cTF
4051     { \__xeCJK_family_csname:n {#1} }
4052     { \prg_return_true: }
4053     { \prg_return_false: }
4054   }
4055 }
4056 \prg_generate_conditional_variant:Nnn \xeCJK_family_if_exist:n { e } { T , F , TF }

```

\CJKfamily 用于切换 CJK 字体族。

```

4057 \NewDocumentCommand \CJKfamily { t+ t- m }
4058 {
4059   \xeCJK_family:NNe #1 #2 {#3}
4060   \tex_ignorespaces:D
4061 }
4062 \cs_new_protected:Npn \xeCJK_family:NNn #1#2#3
4063 {
4064   \tl_if_blank:nTF {#3}
4065   {
4066     \bool_if:NF #1 { \bool_if:NF #2 { \use_none:nn } }
4067     \xeCJK_family_if_exist_use:e { \l_xeCJK_family_tl }
4068   }
4069   {
4070     \bool_if:NTF #2
4071     { \xeCJK_family_if_exist_use:n {#3} }
4072     {
4073       \xeCJK_family_if_exist:nTF {#3}
4074       {
4075         \tl_set:Nn \l_xeCJK_family_tl {#3}
4076         \tl_set_eq:NN \CJK@family \l__xeCJK_fontspec_family_tl
4077         \bool_if:NT #1 { \__xeCJK_family_use:n {#3} }
4078       }
4079       { \__xeCJK_family_unknown_warning:n {#3} }
4080     }
4081   }
4082 }
4083 \cs_generate_variant:Nn \xeCJK_family:NNn { NNe }
4084 \cs_new_protected:Npn \xeCJK_switch_family:n #1
4085 {
4086   \xeCJK_family_if_exist:nTF {#1}
4087   {
4088     \tl_set:Nn \l_xeCJK_family_tl {#1}
4089     \tl_set_eq:NN \CJK@family \l__xeCJK_fontspec_family_tl
4090   }
4091   { \__xeCJK_family_unknown_warning:n {#1} }
4092 }
4093 \cs_generate_variant:Nn \xeCJK_switch_family:n { e , o }

```

PunctFamily 设置汉字标点符号的字体。

```

4094 \keys_define:nn { xeCJK / options }
4095 {

```

```

4096 PunctFamily .choice: ,
4097 PunctFamily .value_required:n = { true } ,
4098 PunctFamily / false .code:n =
4099 {
4100     \tl_set:Nn \CJK@punctfamily { \CJK@family }
4101     \tl_set:Nn \l_xeCJK_punct_family_tl { \l_xeCJK_family_tl }
4102     \xeCJK_cs_clear:N \__xeCJK_select_font:
4103     \xeCJK_cs_clear:N \__xeCJK_select_punct_font:
4104     \cs_set_eq:NN \xeCJK_select_punct_font: \xeCJK_select_font:
4105 } ,
4106 PunctFamily / unknown .code:n =
4107 { \xeCJK_punct_family:e {#1} } ,
4108 }
4109 \cs_new_protected:Npn \xeCJK_punct_family:n #1
4110 {
4111     \xeCJK_family_if_exist:nTF {#1}
4112     {
4113         \tl_set:Nn \l_xeCJK_punct_family_tl {#1}
4114         \tl_set_eq:NN \CJK@punctfamily \l__xeCJK_fontspec_family_tl
4115         \cs_set_eq:NN \__xeCJK_select_font: \xeCJK_select_font:
4116         \cs_set_eq:NN \__xeCJK_select_punct_font: \__xeCJK_select_punct_font_aux:
4117         \cs_set_eq:NN \xeCJK_select_punct_font: \__xeCJK_select_punct_font:
4118     }
4119     { \__xeCJK_family_unknown_warning:n {#1} }
4120 }
4121 \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。

```
4122 \tl_new:N \l_xeCJK_family_tl
```

\CJK@family 用于保存实际的字体族名称。

```
4123 \tl_new:N \CJK@family
```

```

\__xeCJK_gobble_CJKfamily: 4124 \cs_new_protected:Npn \__xeCJK_gobble_CJKfamily:
4125 { \cs_set_eq:NN \CJKfamily \__xeCJK_gobble_CJKfamily:wn }
4126 \NewExpandableDocumentCommand \__xeCJK_gobble_CJKfamily:wn { t+ t- m } { }

```

```

\xeCJK_family_if_exist_use:n 4127 \cs_new_protected:Npn \xeCJK_family_if_exist_use:n #1
4128 {
4129     \xeCJK_family_if_exist:nTF {#1}
4130     { \__xeCJK_family_use:n {#1} }
4131     { \__xeCJK_family_unknown_warning:n {#1} }
4132 }
4133 \cs_generate_variant:Nn \xeCJK_family_if_exist_use:n { e }

```

```

\__xeCJK_family_unknown_warning:n 4134 \cs_new_protected:Npn \__xeCJK_family_unknown_warning:n #1
4135 {
4136     \prop_if_empty:NF \g__xeCJK_family_font_name_prop
4137     {
4138         \seq_if_in:NnF \g__xeCJK_unknown_family_seq {#1}
4139         {
4140             \seq_gput_right:Nn \g__xeCJK_unknown_family_seq {#1}
4141             \__xeCJK_warning:ne { CJKfamily-Unknown } {#1}
4142         }
4143     }
4144 }
4145 \seq_new:N \g__xeCJK_unknown_family_seq
4146 \__xeCJK_msg_new:nn { CJKfamily-Unknown }
4147 {
4148     Unknown~CJK~family~\__xeCJK_msg_family_map:n {#1}'~is~being~ignored.\\
4149     Try~to~use~\__xeCJK_msg_def_family_map:n {#1}'~to~define~it.
4150 }
4151 \cs_new:Npn \__xeCJK_msg_def_family_map:n #1
4152 {
4153     \str_case_e:nnF {#1}
4154     {
4155         \CJKrmdefault { \token_to_str:N \setCJKmainfont }

```

```

4156         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
4157         \CJKttdefault { \token_to_str:N \setCJKmonofont }
4158     }
4159     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
4160     [<...>]\{<...>\}
4161 }
4162 \cs_new:Npn \__xeCJK_msg_family_map:n #1
4163 {
4164     \str_case:e:nnF {#1}
4165     {
4166         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
4167         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
4168         \CJKttdefault { \token_to_str:N \CJKttdefault }
4169     }
4170     {#1}
4171 }

```

`\__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。

```

4172 \cs_new_protected:Npn \__xeCJK_pass_args:nnnn #1#2#3#4
4173 {
4174     \tl_if_novalue:nTF {#2}
4175     { \__xeCJK_post_arg:w {#1} {#3} {#4} }
4176     {
4177         \use:e { #1 {#2} {#3} }
4178         #4
4179     }
4180 }
4181 \NewDocumentCommand \__xeCJK_post_arg:w { m m m O { } }
4182 {
4183     \use:e { #1 {#4} {#2} }
4184     #3
4185 }

```

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

```

\setCJKsansfont
\setCJKmonofont
4186 \NewDocumentCommand \setCJKmainfont { o m }
4187 {
4188     \__xeCJK_pass_args:nnnn
4189     { \xeCJK_set_family:nnn { \CJKrmdefault } } {#1} {#2}
4190     { \__xeCJK_preamble_family:n { \CJKrmdefault } }
4191 }
4192 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
4193 \NewDocumentCommand \setCJKsansfont { o m }
4194 {
4195     \__xeCJK_pass_args:nnnn
4196     { \xeCJK_set_family:nnn { \CJKsfdefault } } {#1} {#2}
4197     { \__xeCJK_preamble_family:n { \CJKsfdefault } }
4198 }
4199 \NewDocumentCommand \setCJKmonofont { o m }
4200 {
4201     \__xeCJK_pass_args:nnnn
4202     { \xeCJK_set_family:nnn { \CJKttdefault } } {#1} {#2}
4203     { \__xeCJK_preamble_family:n { \CJKttdefault } }
4204 }
4205 \@onlypreamble \setCJKmainfont
4206 \@onlypreamble \setCJKmathfont
4207 \@onlypreamble \setCJKsansfont
4208 \@onlypreamble \setCJKmonofont
4209 \@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。

```

4210 \cs_new_protected:Npn \__xeCJK_preamble_family:n #1
4211 { \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
4212 \NewDocumentCommand \setCJKfamilyfont { m o m }
4213 {
4214   \__xeCJK_pass_args:nnnn
4215   { \xeCJK_set_family:nnn {#1} } {#2} {#3}
4216   { }
4217 }
4218 \__xeCJK_msg_new:nn { command-already-defined }
4219 { Control~sequence~#1~is~already~defined. }
4220 \NewDocumentCommand \newCJKfontfamily { o m o m }
4221 {
4222   \tl_set:Np \l__xeCJK_tmp_tl
4223   { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
4224   \cs_if_exist:NTF #2
4225   { \__xeCJK_error:ne { command-already-defined } { \token_to_str:N #2 } }
4226   {
4227     \cs_set_protected:Npe #2
4228     { \xeCJK_switch_family:n { \l__xeCJK_tmp_tl } }
4229   }
4230   \__xeCJK_pass_args:nnnn
4231   { \xeCJK_set_family:nnn { \l__xeCJK_tmp_tl } } {#3} {#4}
4232   { }
4233 }
4234 \NewDocumentCommand \CJKfontspec { o m }
4235 {
4236   \__xeCJK_pass_args:nnnn
4237   { \xeCJK_fontspec:nn } {#1} {#2}
4238   { \tex_ignorespaces:D }
4239 }
4240 \xeCJK_fontspec:nn \cs_new_protected:Npn \xeCJK_fontspec:nn #1#2
4241 {
4242   \prop_get:NnNTF \g__xeCJK_fontspec_prop
4243   { CJKfontspec/#1/#2/id } \l_xeCJK_family_tl
4244   { \xeCJK_switch_family:o { \l_xeCJK_family_tl } }
4245   {
4246     \__xeCJK_fontspec:enn
4247     { CJKfontspec ( \int_eval:n { \g__xeCJK_family_int + 1 } ) }
4248     {#1} {#2}
4249   }
4250 }
4251 \cs_new_protected:Npn \__xeCJK_fontspec:nnn #1#2#3
4252 {
4253   \prop_gput:Nnn \g__xeCJK_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
4254   \xeCJK_set_family:nnn {#1} {#2} {#3}
4255   \xeCJK_switch_family:n {#1}
4256 }
4257 \cs_generate_variant:Nn \xeCJK_fontspec:nn { oo }
4258 \cs_generate_variant:Nn \__xeCJK_fontspec:nnn { e }
4259 \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
4260 \clist_new:N \g__xeCJK_default_features_clist
4261 \NewDocumentCommand \defaultCJKfontfeatures { m }
4262 { \clist_gset:Nn \g__xeCJK_default_features_clist {#1} }
4263 \@onlypreamble \defaultCJKfontfeatures
4264 \NewDocumentCommand \addCJKfontfeatures { s o { } m }
4265 {
4266   \xeCJK_add_font_features:Nee #1 {#2} {#3}
4267   \tex_ignorespaces:D
4268 }
4269 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
4270 \xeCJK_add_font_features:Nnn \cs_new_protected:Npn \xeCJK_add_font_features:Nnn #1#2#3
4271 {
4272   \prop_get:NnNTF \g__xeCJK_family_font_name_prop
4273   \l_xeCJK_family_tl \l__xeCJK_font_name_tl

```

```

4274 {
4275   \clist_set:Nn \l__xeCJK_add_font_features_clist {#3}
4276   \seq_map_inline:Nn \g__xeCJK_sub_key_seq
4277   { \clist_remove_all:Nn \l__xeCJK_add_font_features_clist {##1} }
4278   \seq_clear:N \l__xeCJK_sub_key_seq
4279   \clist_clear:N \l__xeCJK_add_block_features_clist
4280   \clist_map_function:nN {#2} \__xeCJK_add_sub_features:n
4281   \bool_lazy_and:nnT
4282   {#1}
4283   { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
4284   {
4285     \seq_map_function:NN
4286     \g__xeCJK_sub_key_seq \__xeCJK_add_sub_class_features:n
4287   }
4288   \prop_get:NoNT \g__xeCJK_family_font_options_prop
4289   \l_xeCJK_family_tl \l__xeCJK_font_options_clist
4290   {
4291     \bool_lazy_or:nnT
4292     { \seq_if_empty_p:N \l__xeCJK_sub_key_seq }
4293     {#1}
4294     {
4295       \clist_concat:NNN \l__xeCJK_font_options_clist
4296       \l__xeCJK_font_options_clist \l__xeCJK_add_font_features_clist
4297     }
4298     \clist_concat:NNN \l__xeCJK_font_options_clist
4299     \l__xeCJK_font_options_clist \l__xeCJK_add_block_features_clist
4300   }
4301   \xeCJK_fontspec:oo \l__xeCJK_font_options_clist \l__xeCJK_font_name_tl
4302   }
4303   { \__xeCJK_warning:n { addCJKfontfeature-ignored } }
4304   }
4305   \cs_new_protected:Npn \__xeCJK_add_sub_features:n #1
4306   {
4307     \seq_if_in:NnTF \g__xeCJK_sub_key_seq {#1}
4308     {
4309       \seq_put_right:Nn \l__xeCJK_sub_key_seq {#1}
4310       \__xeCJK_add_sub_class_features:n {#1}
4311     }
4312     { \__xeCJK_warning:ne { SubBlock-undefined } {#1} }
4313   }
4314   \clist_new:N \l__xeCJK_add_font_features_clist
4315   \clist_new:N \l__xeCJK_add_block_features_clist
4316   \cs_generate_variant:Nn \xeCJK_add_font_features:Nnn { Nee , Nne }
4317   \__xeCJK_msg_new:nn { addCJKfontfeature-ignored }
4318   {
4319     \token_to_str:N \addCJKfontfeature (s)~ignored.\\\
4320     It~cannot~be~used~with~a~font~that~wasn't~selected~by~xeCJK.
4321   }
4322   \cs_new_protected:Npn \__xeCJK_add_sub_class_features:n #1
4323   {
4324     \prop_get:NoNTF \g__xeCJK_family_font_name_prop
4325     { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_name_tl
4326     {
4327       \prop_get:NoN \g__xeCJK_family_font_options_prop
4328       { \l_xeCJK_family_tl/#1 } \l__xeCJK_sub_font_options_clist
4329     }
4330     {
4331       \prop_get:NeNTF \g__xeCJK_family_font_name_prop
4332       { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_name_tl
4333       {
4334         \prop_get:NeN \g__xeCJK_family_font_options_prop
4335         { \CJKfamilydefault/#1 } \l__xeCJK_sub_font_options_clist
4336       }
4337       {
4338         \prop_get:NoN \g__xeCJK_family_font_options_prop
4339         \l_xeCJK_family_tl \l__xeCJK_sub_font_options_clist
4340         \tl_set_eq:NN \l__xeCJK_sub_font_name_tl \l__xeCJK_font_name_tl

```

```

4341     }
4342   }
4343   \clist_concat:NNN \l__xeCJK_sub_font_options_clist
4344   \l__xeCJK_sub_font_options_clist \l__xeCJK_add_font_features_clist
4345   \clist_put_right:Ne \l__xeCJK_add_block_features_clist
4346   {
4347     #1 =
4348     {
4349       [ \exp_not:o \l__xeCJK_sub_font_options_clist ]
4350       { \exp_not:o \l__xeCJK_sub_font_name_tl }
4351     }
4352   }
4353 }
4354 \cs_generate_variant:Nn \prop_get:NnN { Ne }
4355 \prg_generate_conditional_variant:Nnn \prop_get:NnN { Ne } { T , TF }

LoadFandol 4356 \keys_define:nn { xeCJK / options }
4357   { LoadFandol .bool_gset:N = \g__xeCJK_fandol_bool }
4358 \cs_new_protected:Npn \__xeCJK_load_fandol:
4359   {
4360     \xeCJK_set_family:enn { \CJKrmdefault }
4361     { Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular }
4362     { FandolSong-Regular }
4363     \xeCJK_set_family:enn { \CJKsfdefault }
4364     { Extension = .otf , BoldFont = FandolHei-Bold } { FandolHei-Regular }
4365     \xeCJK_set_family:enn { \CJKttdefault }
4366     { Extension = .otf } { FandolFang-Regular }
4367   }

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。

4368 \__xeCJK_at_end_preamble:n
4369   {
4370     \tl_if_eq:NNT \CJKfamilydefault \l__xeCJK_family_default_init_tl
4371     {
4372       \group_begin:
4373       \cs_set_eq:NN \__xeCJK_family_default_wrap:n \exp_not:n
4374       \tl_gset:Ne \CJKfamilydefault
4375       {
4376         \str_case:onF { \familydefault }
4377         {
4378           { \rmdefault } { \exp_not:N \CJKrmdefault }
4379           { \sfdefault } { \exp_not:N \CJKsfdefault }
4380           { \ttdefault } { \exp_not:N \CJKttdefault }
4381         }
4382         { \CJKfamilydefault }
4383       }
4384       \group_end:
4385     }
4386     \prop_if_empty:NTF \g__xeCJK_family_font_name_prop
4387     {
4388       \bool_if:NTF \g__xeCJK_fandol_bool
4389       {
4390         \__xeCJK_warning:n { fandol }
4391         \__xeCJK_load_fandol:
4392         \xeCJK_ensure_default_family:
4393       }
4394       { \__xeCJK_warning:ne { no-CJKfamily } { \CJKfamilydefault } }
4395     }
4396     { \xeCJK_ensure_default_family: }
4397   }
4398 \cs_new_protected:Npn \xeCJK_ensure_default_family:
4399   {
4400     \xeCJK_family_if_exist:eF { \CJKfamilydefault }

```

```

4401 {
4402   \tl_set_eq:NN \l__xeCJK_tmp_tl \CJKfamilydefault
4403   \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
4404   { \use:n }
4405   {
4406     \xeCJK_family_if_exist:eTF { \CJKrmdefault }
4407     { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } }
4408   }
4409   {
4410     \prop_map_inline:Nn \g__xeCJK_family_font_name_prop
4411     {
4412       \prop_map_break:n
4413       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
4414     }
4415   }
4416   \__xeCJK_warning:nee { CJKfamilydefault-undefined }
4417   { \l__xeCJK_tmp_tl } { \CJKfamilydefault }
4418 }
4419 \xeCJK_switch_family:e { \CJKfamilydefault }
4420 \bool_if:NT \g__xeCJK_math_bool { \xeCJK_set_mathfont: }
4421 }
4422 \__xeCJK_msg_new:nn { no-CJKfamily }
4423 {
4424   It~seems~that~you~have~not~declared~a~CJKfamily.\\
4425   If~you~want~to~use~xeCJK~in~the~right~way,~you~should~use\\\\
4426   `\_xeCJK_msg_def_family_map:n {#1}'\\\\
4427   in~the~preamble~to~declare~the~default~CJKfamily.\\
4428 }
4429 \__xeCJK_msg_new:nn { CJKfamilydefault-undefined }
4430 {
4431   Undefined~CJK~default~family~`\_xeCJK_msg_family_map:n {#1}'~
4432   has~been~replaced~by~`\_xeCJK_msg_family_map:n {#2}'\\.\\\\
4433   Try~to~use~`\_xeCJK_msg_def_family_map:n {#1}'~to~define~it.
4434 }
4435 \__xeCJK_msg_new:nn { fandol }
4436 {
4437   Fandol~is~being~set~as~the~default~font~for~CJK~text.\\
4438   Please~make~sure~it~has~been~properly~installed.
4439 }

```

5.14 数学字体设置

CJKmath 是否启用 CJK 数学字体的宏包选项。

```
4440 \keys_define:nn { xeCJK / options } { CJKmath .bool_gset:N = \g__xeCJK_math_bool }
```

\setCJKmathfont 设置 CJK 数学字体。

```

4441 \NewDocumentCommand \setCJKmathfont { o m }
4442 {
4443   \__xeCJK_pass_args:nnnn
4444   { \xeCJK_set_family:nnn { \c__xeCJK_math_tl } } { #1 } { #2 }
4445   { }
4446 }
4447 \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。

```

4448 \cs_new_protected:Npn \xeCJK_set_mathfont:
4449 {
4450   \cs_if_exist_use:N \__xeCJK_save_um_char:
4451   \xeCJK_family_if_exist:eTF { \c__xeCJK_math_tl }
4452   { \__xeCJK_set_mathfont_aux: }
4453   {
4454     \xeCJK_family_if_exist:eT { \CJKfamilydefault }
4455     {
4456       \__xeCJK_copy_family:ee { \c__xeCJK_math_tl } { \CJKfamilydefault }

```

```

4457         \__xeCJK_set_mathfont_aux:
4458     }
4459 }
4460 \cs_if_exist_use:N \__xeCJK_restore_um_char:
4461 }
4462 \cs_new_protected:Npn \__xeCJK_set_mathfont_aux:
4463 {
4464     \tl_const:Nc \c__xeCJK_math_family_tl
4465     { \l__xeCJK_fontspec_family_tl }
4466     \xeCJK_declare_mathfont:ee
4467     { \c__xeCJK_math_tl }
4468     { \c__xeCJK_math_family_tl }
4469     \int_const:Nn \c_xeCJK_math_fam_int
4470     { \use:c { sym \c__xeCJK_math_tl } }
4471     \clist_gconcat:NNN \g__xeCJK_math_chars_clist
4472     \g__xeCJK_CJK_range_clist \g__xeCJK_FullLeft_range_clist
4473     \clist_gconcat:NNN \g__xeCJK_math_chars_clist
4474     \g__xeCJK_math_chars_clist \g__xeCJK_FullRight_range_clist
4475     \xeCJK_gset_mathcode:Nn \g__xeCJK_math_chars_clist
4476     { \c_xeCJK_math_fam_int }
4477     \xeCJK_set_mathfont_block:
4478 }
4479 \clist_new:N \g__xeCJK_math_chars_clist
4480 \prop_new:N \g__xeCJK_fam_prop

```

\xeCJK_set_mathfont_block: 分区数学字体。

```

4481 \cs_new_protected:Npn \xeCJK_set_mathfont_block:
4482 {
4483     \seq_if_empty:NF \g__xeCJK_CJK_sub_class_seq
4484     {
4485         \seq_map_function:NN
4486         \g__xeCJK_CJK_sub_class_seq
4487         \xeCJK_set_mathfont_block:n
4488     }
4489 }
4490 \cs_new_protected:Npn \xeCJK_set_mathfont_block:n #1
4491 {
4492     \xeCJK_block_family:nn { \c__xeCJK_math_tl } {#1}
4493     \prop_get:NoNTF \g__xeCJK_fam_prop
4494     \l__xeCJK_fontspec_family_tl \l__xeCJK_tmp_tl
4495     { \int_set:Nn \l__xeCJK_fam_int { \l__xeCJK_tmp_tl } }
4496     {
4497         \xeCJK_declare_mathfont:ee
4498         { \c__xeCJK_math_tl / #1 }
4499         { \l__xeCJK_fontspec_family_tl }
4500         \__xeCJK_set_mathfont_block_aux:cn
4501         { sym \c__xeCJK_math_tl / #1 } {#1}
4502     }
4503     \xeCJK_gset_mathcode:cn { g__xeCJK_CJK/#1_range_clist } { \l__xeCJK_fam_int }
4504 }
4505 \cs_new_protected:Npn \__xeCJK_set_mathfont_block_aux:Nn #1#2
4506 {
4507     \int_set_eq:NN \l__xeCJK_fam_int #1
4508     \prop_gput:Nnn \g__xeCJK_block_fam_prop {#2} {#1}
4509 }
4510 \int_new:N \l__xeCJK_fam_int
4511 \prop_new:N \g__xeCJK_block_fam_prop
4512 \cs_generate_variant:Nn \__xeCJK_set_mathfont_block_aux:Nn { c }

```

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

```

4513 \cs_new_protected:Npn \xeCJK_declare_mathfont:nn #1#2
4514 {
4515     \xeCJK_declare_symbol_font:nnnn {#1} { \c__xeCJK_encoding_tl }

```

```

4516     {#2} { \mddefault } { \shapedefault }
4517     \cs_if_free:cF
4518     { \c__xeCJK_encoding_tl/#2/\bfdefault/\shapedefault }
4519     {
4520         \SetSymbolFont {#1} { bold } { \c__xeCJK_encoding_tl }
4521         {#2} { \bfdefault } { \shapedefault }
4522     }
4523     \prop_gput:Nne \g__xeCJK_fam_prop {#2} { \exp_not:c { sym #1 } }
4524 }
4525 \cs_generate_variant:Nn \prop_put:Nnn { Nne }
4526 \cs_generate_variant:Nn \prop_gput:Nnn { Nne }
4527 \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。

```

4528 \cs_new_protected:Npn \xeCJK_declare_symbol_font:nnnnn #1
4529 { \__xeCJK_declare_symbol_font:cnnnn { sym #1 } }
4530 \cs_new_protected:Npn \__xeCJK_declare_symbol_font:Nnnnn #1
4531 {
4532     \xeCJK_new_fam:N #1
4533     \xeCJK_new_symbol_font:Nnnnn #1
4534 }
4535 \cs_generate_variant:Nn \__xeCJK_declare_symbol_font:Nnnnn { c }

```

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

```

4536 \cs_new_protected:Npn \xeCJK_new_fam:N #1
4537 {
4538     \int_compare:nNnTF
4539     { \g__xeCJK_fam_allocation_int } > { \g__xeCJK_fam_bottom_int }
4540     {
4541         \int_set_eq:NN \allocationnumber \g__xeCJK_fam_allocation_int
4542         \int_const:Nn #1 { \allocationnumber }
4543         \iow_log:e
4544         {
4545             \token_to_str:N #1 =
4546             \token_to_str:N \mathgroup \int_use:N \allocationnumber
4547         }
4548         \int_gdecr:N \g__xeCJK_fam_allocation_int
4549     }
4550     { \__xeCJK_error:n { fam-exhausted } }
4551 }
4552 \tex_countdef:D \g__xeCJK_fam_bottom_int = 18 ~
4553 \int_new:N \g__xeCJK_fam_allocation_int
4554 \int_gset:Nn \g__xeCJK_fam_allocation_int { 255 }
4555 \__xeCJK_msg_new:nn { fam-exhausted }
4556 { No~room~for~a~new~fam. }

```

\xeCJK_new_symbol_font:Nnnnnn 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
4557 \cs_new_protected:Npn \xeCJK_new_symbol_font:Nnnnn #1#2#3#4#5
4558 { \__xeCJK_new_symbol_font:Nc #1 { #2/#3/#4/#5 } }
4559 \cs_new_protected:Npn \__xeCJK_new_symbol_font:NN #1#2
4560 {
4561     \tl_put_right:Nn \group@list { \group@elt #1 #2 }
4562     \cs_set:Npn \version@elt ##1
4563     { \tl_put_right:Nn ##1 { \getanddefine@fonts #1 #2 } }
4564     \version@list
4565 }
4566 \cs_generate_variant:Nn \__xeCJK_new_symbol_font:NN { Nc }

```

CJK 字符的数学类别固定为 0(\mathord)。

```

\xeCJK_gset_mathcode:Nn
\xeCJK_gset_mathcode:Nnn
\xeCJK_gset_mathcode:nnnn
4567 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nn #1#2
4568 {
4569     \clist_map_inline:Nn #1

```

```

4570     {
4571         \__xeCJK_set_char_class_aux:Nnw \xeCJK_gset_mathcode:nnnn { ##1 }
4572         { 0 } {#2}
4573     }
4574 }
4575 \cs_generate_variant:Nn \xeCJK_gset_mathcode:Nn { c }
4576 \cs_new_protected:Npn \xeCJK_gset_mathcode:nnnn #1#2#3#4
4577 {
4578     \__xeCJK_check_num_range:nnNN {#1} {#2} \l__xeCJK_begin_int \l__xeCJK_end_int
4579     \xeCJK_int_until_do:nn { \l__xeCJK_begin_int > \l__xeCJK_end_int }
4580     {
4581         \xeCJK_gset_mathcode:Nnn \l__xeCJK_begin_int {#3} {#4}
4582         \int_incr:N \l__xeCJK_begin_int
4583     }
4584 }
4585 \cs_new_protected:Npn \xeCJK_gset_mathcode:Nnn #1#2#3
4586 { \tex_global:D \tex_Umathcode:D #1 = #2 ~ #3 ~ #1 }

```

5.15 抄录环境中的间距调整

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

```

4587 \int_new:N \l__xeCJK_verb_case_int
4588 \keys_define:nn { xeCJK / options }
4589 {
4590     Verb .choices:nn =
4591         { true , env+ , env , false }
4592         { \int_set_eq:NN \l__xeCJK_verb_case_int \l_keys_choice_int } ,
4593     Verb .default:n = { env }
4594 }
4595 \cs_new_protected:Npn \__xeCJK_verb_font_hook:
4596 {
4597     \if_case:w \l__xeCJK_verb_case_int
4598     \or:
4599         \__xeCJK_nobreak_skip_zero:
4600     \or:
4601         \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
4602         { \xeCJKVerbAddon }
4603         { \__xeCJK_nobreak_skip: }
4604     \or:
4605         \int_compare:nNnTF \tex_currentgrouptype:D = { 14 }
4606         { \xeCJKVerbAddon }
4607         { \__xeCJK_nobreak_skip_zero: }
4608     \fi:
4609 }
4610 \__xeCJK_after_preamble:n
4611 {
4612     \cs_set_protected:Npe \verbatim@font
4613     { \exp_not:o { \verbatim@font } \__xeCJK_verb_font_hook: }
4614 }

```

```

\__xeCJK_nobreak_skip_zero: 4615 \cs_new_protected:Npn \__xeCJK_nobreak_skip_zero:
\__xeCJK_nobreak_skip: 4616 {
4617     \__xeCJK_reset_shipout_skip:
4618     \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
4619     \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
4620     \tl_put_right:Nn \l__xeCJK_reset_shipout_skip_hook_tl
4621     {
4622         \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
4623         \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
4624     }
4625     \xeCJK_cs_clear:N \CJKglue
4626     \xeCJK_cs_clear:N \CJKecglue
4627     \xeCJK_cs_clear:N \xeCJK_check_for_glue:
4628     \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \xeCJK_class_group_end:

```

```

4629 \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
4630 \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
4631 }
4632 \cs_new_protected:Npn \__xeCJK_nobreak_skip:
4633 {
4634   \__xeCJK_reset_shipout_skip:
4635   \xeCJK_glue_to_skip:nN { \CJKglue } \l__xeCJK_ccglue_skip
4636   \skip_if_eq:nnTF { \l__xeCJK_ccglue_skip } { \c_zero_skip }
4637     { \xeCJK_cs_clear:N \CJKglue }
4638     { \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue: }
4639   \xeCJK_glue_to_skip:nN { \CJKecglue } \l__xeCJK_ecglue_skip
4640   \skip_if_eq:nnTF { \l__xeCJK_ecglue_skip } { \c_zero_skip }
4641     { \xeCJK_cs_clear:N \CJKecglue }
4642     { \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue: }
4643   \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_nobreak_hskip:n
4644   \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n \__xeCJK_nobreak_hskip:n
4645 }
4646 \cs_new_protected:Npn \__xeCJK_nobreak_ccglue:
4647 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ccglue_skip }
4648 \cs_new_protected:Npn \__xeCJK_nobreak_ecglue:
4649 { \xeCJK_no_break: \skip_horizontal:N \l__xeCJK_ecglue_skip }

\__xeCJK_reset_shipout_skip: 4650 \cs_new_protected:Npn \__xeCJK_reset_shipout_skip:
4651 {
4652   \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
4653   \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
4654   \cs_set_eq:NN \__xeCJK_shipout_punct_hskip:n \__xeCJK_punct_hskip:n
4655   \cs_set_eq:NN
4656     \__xeCJK_shipout_punct_breakable_kern:n \__xeCJK_punct_breakable_kern:n
4657   \tl_set:N \l__xeCJK_off_verb_addon_tl
4658   {
4659     \bool_if:NTF \l__xeCJK_xecglue_bool
4660       { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
4661       { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
4662     \exp_not:n
4663     {
4664       \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
4665       \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
4666       \cs_set_eq:NN \__xeCJK_punct_hskip:n \__xeCJK_shipout_punct_hskip:n
4667       \cs_set_eq:NN \__xeCJK_punct_breakable_kern:n
4668         \__xeCJK_shipout_punct_breakable_kern:n
4669       \l__xeCJK_reset_shipout_skip_hook_tl
4670     }
4671   }
4672   \xeCJK_add_to_shipout:n { \l__xeCJK_off_verb_addon_tl }
4673   \keys_set:nn { xeCJK / options } { xCJKecglue = false }
4674 }
4675 \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。

```

4676 \NewDocumentCommand \xeCJKVerbAddOn { }
4677 {
4678   \int_compare:nNnF \tex_currentgrouplevel:D = \c_zero_int
4679   {
4680     \bool_if:NF \l__xeCJK_listings_env_bool
4681     {
4682       \dim_compare:nNnTF
4683         { \tex_fontdimen:D 2 ~ \tex_font:D } =
4684         { \tex_fontcharwd:D \tex_font:D \c__xeCJK_mono_letter_int }
4685         {
4686           \__xeCJK_set_verb_exspace:
4687           \__xeCJK_verb_addon:
4688         }
4689     }
4690   }

```

```

4689         {
4690             \int_if_odd:nTF { \l__xeCJK_verb_case_int }
4691             { \__xeCJK_nobreak_skip_zero: }
4692             { \__xeCJK_nobreak_skip: }
4693         }
4694     }
4695 }
4696 }
4697 \int_const:Nn \c__xeCJK_mono_letter_int { 77 }
4698 \bool_new:N \l__xeCJK_listings_env_bool
4699 \NewDocumentCommand \xeCJKOffVerbAddon { }
4700 { \tl_use:N \l__xeCJK_off_verb_addon_tl }
4701 \tl_new:N \l__xeCJK_off_verb_addon_tl
4702 \cs_new_protected:Npn \__xeCJK_verb_addon:
4703 {
4704     \bool_if:NF \l__xeCJK_verb_addon_bool
4705     { \__xeCJK_verb_addon_action: }
4706     \skip_if_eq:nnTF { \l__xeCJK_verb_exspace_skip } { \c_zero_skip }
4707     {
4708         \xeCJK_cs_clear:N \CJKglue
4709         \xeCJK_cs_clear:N \CJKecglue
4710     }
4711     {
4712         \skip_set_eq:NN \l__xeCJK_ccglue_skip \l__xeCJK_verb_exspace_skip
4713         \skip_set:Nn \l__xeCJK_ecglue_skip { \l__xeCJK_verb_exspace_skip / 2 }
4714         \cs_set_eq:NN \CJKglue \__xeCJK_nobreak_ccglue:
4715         \cs_set_eq:NN \CJKecglue \__xeCJK_nobreak_ecglue:
4716     }
4717     \cs_set_eq:NN \xeCJK_check_for_glue: \CJKecglue
4718     \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_verb_CJK_and_Boundary:w
4719 }
4720 \cs_new_protected:Npn \__xeCJK_verb_addon_action:
4721 {
4722     \bool_set_true:N \l__xeCJK_verb_addon_bool
4723     \__xeCJK_set_char_class_eq:nn { FullLeft } { CJK }
4724     \__xeCJK_set_char_class_eq:nn { FullRight } { CJK }
4725     \__xeCJK_set_char_class_eq:nn { HalfLeft } { Default }
4726     \__xeCJK_set_char_class_eq:nn { HalfRight } { Default }
4727     \__xeCJK_set_char_class_eq:nn { NormalSpace } { Default }
4728     \cs_set_eq:NN \__xeCJK_shipout_CJKglue: \CJKglue
4729     \cs_set_eq:NN \__xeCJK_shipout_CJKecglue: \CJKecglue
4730     \cs_set_eq:NN \__xeCJK_shipout_check_for_glue: \xeCJK_check_for_glue:
4731     \cs_set_eq:NN \__xeCJK_shipout_boundary:w \xeCJK_CJK_and_Boundary:w
4732     \set_protected:Npe \xeCJKOffVerbAddon
4733     {
4734         \__xeCJK_reset_char_class:n { FullLeft }
4735         \__xeCJK_reset_char_class:n { FullRight }
4736         \__xeCJK_reset_char_class:n { HalfLeft }
4737         \__xeCJK_reset_char_class:n { HalfRight }
4738         \__xeCJK_reset_char_class:n { NormalSpace }
4739         \bool_if:NTF \l__xeCJK_xecglue_bool
4740         { \keys_set:nn { xeCJK / options } { xCJKecglue = true } }
4741         { \keys_set:nn { xeCJK / options } { xCJKecglue = false } }
4742         \exp_not:n
4743         {
4744             \cs_set_eq:NN \CJKglue \__xeCJK_shipout_CJKglue:
4745             \cs_set_eq:NN \CJKecglue \__xeCJK_shipout_CJKecglue:
4746             \cs_set_eq:NN \xeCJK_check_for_glue: \__xeCJK_shipout_check_for_glue:
4747             \cs_set_eq:NN \xeCJK_CJK_and_Boundary:w \__xeCJK_shipout_boundary:w
4748         }
4749     }
4750     \xeCJK_add_to_shipout:n { \xeCJKOffVerbAddon }
4751     \keys_set:nn { xeCJK / options } { xCJKecglue = false }
4752 }
4753 \cs_new_protected:Npn \__xeCJK_verb_CJK_and_Boundary:w
4754 { \xeCJK_class_group_end: \CJKecglue }
4755 \cs_new_protected:Npn \__xeCJK_reset_char_class:n #1

```

```

4756 {
4757   \int_set:Nn \l__xeCJK_tmp_int { \xeCJK_class_num:n {#1} }
4758   \clist_map_inline:cn { c__xeCJK_#1_chars_clist }
4759   { \tex_XeTeXcharclass:D ##1 = \l__xeCJK_tmp_int }
4760 }
4761 \bool_new:N \l__xeCJK_verb_addon_bool
4762 \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。

4763 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:
4764 {
4765   \tl_if_exist:cTF { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
4766   {
4767     \skip_set:Nn \l__xeCJK_verb_exspace_skip
4768     { \use:c { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size } }
4769   }
4770   {
4771     \tl_set:Nn \l__xeCJK_current_coor_tl { \CJK@family/\curr@fontshape }
4772     \prop_get:NnTF \g__xeCJK_scale_family_prop
4773     \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
4774     {
4775       \xeCJK_switch_family:o { \l_xeCJK_family_tl }
4776       \skip_zero:N \l__xeCJK_verb_exspace_skip
4777     }
4778     {
4779       \group_begin: \xeCJK_select_font: \exp_args:NNo \group_end:
4780       \__xeCJK_set_verb_exspace:n
4781       { \dim_use:N \tex_fontcharwd:D \tex_font:D "4E00 ~ }
4782     }
4783   }
4784 }
4785 \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。

```

4786 \cs_new_protected:Npn \__xeCJK_set_verb_exspace:n #1
4787 {
4788   \skip_set:Nn \l__xeCJK_verb_exspace_skip
4789   { 2 \tex_fontdimen:D 2 ~ \tex_font:D - #1 }
4790   \dim_compare:nNnTF \l__xeCJK_verb_exspace_skip < \c_zero_dim
4791   {
4792     \skip_zero:N \l__xeCJK_verb_exspace_skip
4793     \exp_args:Nee \__xeCJK_set_verb_scale:nn
4794     { \dim_to_fp:n { 2 \tex_fontdimen:D 2 ~ \tex_font:D } }
4795     { \dim_to_fp:n {#1} }
4796   }
4797   {
4798     \tl_const:ce { xeCJK/verb/\CJK@family/\curr@fontshape/\f@size }
4799     { \skip_use:N \l__xeCJK_verb_exspace_skip }
4800   }
4801 }

```

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

```

4802 \cs_new_protected:Npn \__xeCJK_set_verb_scale:nn #1#2
4803 {
4804   \fp_set:Nn \l__xeCJK_scale_factor_fp { #1 / #2 }
4805   \__xeCJK_warning:nee { scale-factor }
4806   { \fp_eval:n { trunc ( \l__xeCJK_scale_factor_fp , 4 ) } }
4807   { \fp_eval:n { ceil ( #2 / #1 , 4 ) } }
4808   \xeCJK_add_font_features:Nne \c_true_bool
4809   { } { Scale = { \fp_use:N \l__xeCJK_scale_factor_fp } }
4810   \prop_gput:Noo \g__xeCJK_scale_family_prop
4811   \l__xeCJK_current_coor_tl \l_xeCJK_family_tl
4812 }
4813 \__xeCJK_msg_new:nn { scale-factor }

```

```

4814 {
4815   \token_to_str:N \xeCJKVerbAddon'~may~not~work~properly.\\
4816   You~may~set~Scale=#1'~to~CJKfamily~
4817   \_xeCJK_msg_family_map:n { \l_xeCJK_family_tl },\\
4818   or~set~Scale=#2'~to~family~
4819   \str_if_eq:eeTF \f@family \ttdefault
4820   { \token_to_str:N \ttdefault } { \f@family }'.
4821 }
4822 \fp_new:N \l__xeCJK_scale_factor_fp
4823 \prop_new:N \g__xeCJK_scale_family_prop

```

\xeCJK_visible_space: 如果文档不使用 EU1 作为默认字体编码, 那么默认的打字机字体族很可能是传统的 T_EX 字体,
 \@setupverbvisiblespace 这时可视空格按照 OT1 编码传统一般就是字体中的 \char32。

```

4824 \cs_new_protected:Npn \xeCJK_setup_visible_space:
4825 {
4826   \xeCJK_make_boundary:
4827   \xeCJK_glyph_if_exist:NTF { ~~~~2423 }
4828   { \tl_set:Nn \l__xeCJK_visible_space_tl { ~~~~2423 } }
4829   {
4830     \int_compare:nNnTF { \tex_XeTeXfonttype:D \tex_font:D } = \c_zero_int
4831     {
4832       \tl_set:Nn \l__xeCJK_visible_space_tl
4833       {
4834         \str_if_eq:eeTF { \f@family } { \ttdefault }
4835         { \c_catcode_other_space_tl }
4836         { \exp_not:N \textvisiblespace }
4837       }
4838     }
4839     { \_xeCJK_visible_space_fallback: }
4840   }
4841   \cs_set_eq:NN \@xobeysp \l__xeCJK_visible_space_tl
4842 }
4843 \tl_new:N \l__xeCJK_visible_space_tl
4844 \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。

```

4845 \cs_new_protected:Npn \_xeCJK_visible_space_fallback:
4846 {
4847   \exp_args:Nc \_xeCJK_visible_space_fallback_auxi:N
4848   { xeCJK/space/\curr@fontshape/\f@size }
4849 }
4850 \cs_new_protected:Npn \_xeCJK_visible_space_fallback_auxi:N #1
4851 {
4852   \cs_if_exist:NF #1
4853   { \_xeCJK_visible_space_fallback_auxii:N #1 }
4854   \tl_set:Nn \l__xeCJK_visible_space_tl {#1}
4855 }

```

_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。

```

4856 \cs_new_protected:Npn \_xeCJK_visible_space_fallback_auxii:N #1
4857 {
4858   \group_begin:
4859   \exp_args:No \_xeCJK_set_visible_space_size:n
4860   { \dim_use:N \tex_fontdimen:D 2 ~ \tex_font:D }
4861   \cs_new_protected:Npe #1
4862   { \group_begin: \tex_the:D \tex_font:D ~~~~2423 \group_end: }
4863   \group_end:
4864 }
4865 \cs_new_protected:Npn \_xeCJK_set_visible_space_size:n #1
4866 {
4867   \fontencoding { \UnicodeEncodingName }

```

```

4868 \tl_set:Nn \f@family { lmtt }
4869 \selectfont
4870 \dim_compare:nNnF {#1} = { \tex_fontdimen:D 2 ~ \tex_font:D }
4871 {
4872   \fontsize
4873   {
4874     \dim_eval:n
4875     {
4876       \f@size pt *
4877       \dim_ratio:nn {#1} { \tex_fontdimen:D 2 ~ \tex_font:D }
4878     }
4879   }
4880   { \f@baselineskip }
4881   \selectfont
4882 }
4883 }

```

5.16 xeCJK 其他选项

LocalConfig 声明载入本地配置文件的选项。

```

4884 \keys_define:nn { xeCJK / options }
4885 {
4886   LocalConfig .choice: ,
4887   LocalConfig / false .code:n =
4888   { \bool_gset_false:N \g__xeCJK_config_bool } ,
4889   LocalConfig / true .code:n =
4890   {
4891     \bool_gset_true:N \g__xeCJK_config_bool
4892     \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK }
4893   } ,
4894   LocalConfig / unknown .code:n =
4895   {
4896     \bool_gset_true:N \g__xeCJK_config_bool
4897     \tl_gset:Nn \g__xeCJK_config_name_tl { xeCJK - \l_keys_value_tl }
4898   } ,
4899   LocalConfig .default:n = { true }
4900 }
4901 \tl_new:N \g__xeCJK_config_name_tl
4902 \bool_new:N \g__xeCJK_config_bool

```

CJKnumber 和 **indentfirst** 是过时选项。

```

4903 \keys_define:nn { xeCJK / options }
4904 {
4905   CJKnumber .code:n =
4906   { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { CJKnumb } } ,
4907   indentfirst .code:n =
4908   { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { indentfirst } } ,
4909   normalindentfirst .code:n =
4910   { \__xeCJK_warning:nee { option-deprecated } { \l_keys_key_str } { } }
4911 }
4912 \__xeCJK_msg_new:nn { option-deprecated }
4913 {
4914   The~`#1'~option~is~deprecated.\\
4915   \tl_if_empty:nF {#2}
4916   { You~may~load~the~package~`#2'~after~xeCJK~to~use~its~function.\\ }
4917 }

```

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。

```

4918 \keys_define:nn { xeCJK / options }
4919 {
4920   quiet .code:n =
4921   {
4922     \msg_redirect_module:nnn { xeCJK } { warning } { info }

```

```

4923     \msg_redirect_module:nnn { xeCJK } { info } { none }
4924     \xeCJK_if_package_loaded:nF { fontspec }
4925     { \PassOptionsToPackage { quiet } { fontspec } }
4926   } ,
4927   silent .code:n =
4928   {
4929     \msg_redirect_module:nnn { xeCJK } { warning } { none }
4930     \msg_redirect_module:nnn { xeCJK } { info } { none }
4931     \xeCJK_if_package_loaded:nF { fontspec }
4932     { \PassOptionsToPackage { silent } { fontspec } }
4933   } ,
4934   unknown .code:n =
4935   {
4936     \xeCJK_if_package_loaded:nTF { fontspec }
4937     { \_xeCJK_error:ne { key-unknown } { \l_keys_key_str } }
4938     { \PassOptionsToPackage { \l_keys_key_str } { fontspec } }
4939   }
4940 }
4941 \_xeCJK_msg_new:nn { key-unknown }
4942 {
4943   Sorry,~but~xeCJK/options~does~not~have~a~key~called~`#1'.\\
4944   The~key~`#1'~is~being~ignored.
4945 }

```

5.17 xeCJK 初始化设置

```

\CJKsymbol 4946 \cs_new_eq:NN \CJKsymbol \use:n
\CJKpunctsymbol 4947 \cs_new_eq:NN \CJKpunctsymbol \use:n

```

xeCJK 宏包的初始化设置。

```

4948 \keys_set:nn { xeCJK / options }
4949 {
4950   CJKglue = { \skip_horizontal:n { \c_zero_dim plus 0.08 \tex_baselineskip:D } } ,
4951   CJKecglue = { ~ } ,
4952   xCJKecglue = false ,
4953   CheckSingle = false ,
4954   PlainEquation = false ,
4955   CheckFullRight = false ,
4956   CJKspace = false ,
4957   CJKmath = false ,
4958   xeCJKactive = true ,
4959   LocalConfig = true ,
4960   LoadFandol = true ,
4961   RubberPunctSkip = true ,
4962   Verb = env ,
4963   EmboldenFactor = 4 ,
4964   SlantFactor = 0.167 ,
4965   PunctStyle = quanjiao ,
4966   NewLineCS = { \par \[ } ,
4967   EnvCS = { \begin \end } ,
4968   WidowPenalty = { 10 000 } ,
4969   NoBreakCS = { \footnote \footnotemark \nobreak } ,
4970   KaiMingPunct = { ~~~~3002 ~~~~ff0e ~~~~ff1f ~~~~ff01 } ,
4971   LongPunct = { ~~~~2014 ~~~~2e3a ~~~~2025 ~~~~2026 } ,
4972   NoBreakLongPunct = { ~~~~2025 ~~~~2026 } ,
4973   MiddlePunct = { ~~~~2013 ~~~~2014 ~~~~2e3a ~~~~2027 ~~~~00b7 ~~~~30fb ~~~~ff65 } ,
4974   AllowBreakBetweenPuncts = false
4975 }
4976 \defaultCJKfontfeatures { Script = CJK }

```

半字线连接号¹⁶应为半角宽度。

```

4977 \xeCJKsetwidth { ~~~~2013 } { 0.5 em }

```

执行宏包选项,并载入 fontspec 宏包。

```

4978 \cs_if_exist:NTF \ProcessKeyOptions

```

¹⁶见《夹用英文的中文文本的标点符号用法(草案)》5.13 节。

```

4979 { \ProcessKeyOptions [ xeCJK / options ] }
4980 {
4981   \RequirePackage { l3keys2e }
4982   \ProcessKeysOptions { xeCJK / options }
4983 }
4984 \RequirePackage { fontspec } [ 2020/02/03 ]

```

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

```

4985 \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。

```

4986 \keys_define:nn { xeCJK / options }
4987 {
4988   LocalConfig .code:n =
4989     { \__xeCJK_warning:ne { option-invalid } { \l_keys_key_str } }
4990 }
4991 \__xeCJK_msg_new:nn { option-invalid }
4992 {
4993   The~`#1'~option~can~only~be~set~in~the~optional~argument~to~the\\
4994   \token_to_str:N \usepackage \ command~when~xeCJK~is~being~loaded.\\
4995   Please~do~not~set~it~via~the~\token_to_str:N \xeCJKsetup \ command.
4996 }

```

```

\CJKrmdefault 4997 \tl_if_exist:NF \CJKrmdefault
\CJKsfdefault 4998 { \tl_new:N \CJKrmdefault \tl_gset:Nn \CJKrmdefault { rm } }
\CJKttdefault 4999 \tl_if_exist:NF \CJKsfdefault
\CJKfamilydefault 5000 { \tl_new:N \CJKsfdefault \tl_gset:Nn \CJKsfdefault { sf } }
5001 \tl_if_exist:NF \CJKttdefault
5002 { \tl_new:N \CJKttdefault \tl_gset:Nn \CJKttdefault { tt } }
5003 \tl_new:N \l__xeCJK_family_default_init_tl
5004 \cs_new_eq:NN \__xeCJK_family_default_wrap:n \use:n
5005 \tl_set:Ne \l__xeCJK_family_default_init_tl
5006 {
5007   \exp_not:N \__xeCJK_family_default_wrap:n
5008   {
5009     \tl_if_exist:NTF \CJKfamilydefault
5010       { \exp_not:o \CJKfamilydefault }
5011       { \exp_not:N \CJKrmdefault }
5012   }
5013 }
5014 \tl_if_exist:NF \CJKfamilydefault
5015 { \tl_new:N \CJKfamilydefault }
5016 \tl_gset_eq:NN \CJKfamilydefault \l__xeCJK_family_default_init_tl

```

\xeCJKsetup 在导言区或文档中设置 xeCJK 的接口。

```

5017 \NewDocumentCommand \xeCJKsetup { +m }
5018 {
5019   \keys_set:nn { xeCJK / options } {#1}
5020   \tex_ignorespaces:D
5021 }

```

```

\xeCJKsetemboldenfactor 5022 \NewDocumentCommand \xeCJKsetemboldenfactor { m }
\xeCJKsetslantfactor 5023 { \xeCJKsetup { EmboldenFactor = {#1} } }
5024 \NewDocumentCommand \xeCJKsetslantfactor { m }
5025 { \xeCJKsetup { SlantFactor = {#1} } }

```

```

\punctstyle 5026 \NewDocumentCommand \punctstyle { m } { \xeCJKsetup { PunctStyle = {#1} } }
\xeCJKplainchr 5027 \NewDocumentCommand \xeCJKplainchr { } { \xeCJKsetup { PunctStyle = plain } }

```

```

\CJKsetecglue 5028 \NewDocumentCommand \CJKsetecglue { m } { \xeCJKsetup { CJKecglue = {#1} } }
5029 \cs_new_eq:NN \xeCJKsetecglue \CJKsetecglue

```

```

\CJKspace 5030 \NewDocumentCommand \CJKspace { } { \xeCJKsetup { CJKspace = true } }
\CJKnospace 5031 \NewDocumentCommand \CJKnospace { } { \xeCJKsetup { CJKspace = false } }

```

```

\XeCJKallowbreakbetweenpuncts 5032 \NewDocumentCommand \XeCJKallowbreakbetweenpuncts { }
\XeCJKnobreakbetweenpuncts 5033 { \XeCJKsetup { AllowBreakBetweenPuncts = true } }
5034 \NewDocumentCommand \XeCJKnobreakbetweenpuncts { }
5035 { \XeCJKsetup { AllowBreakBetweenPuncts = false } }

\XeCJKenablefallback 5036 \NewDocumentCommand \XeCJKenablefallback { }
\XeCJKdisablefallback 5037 { \XeCJKsetup { AutoFallBack = true } }
5038 \NewDocumentCommand \XeCJKdisablefallback { }
5039 { \XeCJKsetup { AutoFallBack = false } }

\XeCJKsetcharclass 5040 \_XeCJK_msg_new:nn { XeCJKsetcharclass-deprecated }
5041 {
5042   \token_to_str:N \XeCJKsetcharclass\ is~deprecated~and~
5043   does~not~update~CJKmath~char~lists.\\
5044   Use~\token_to_str:N \XeCJKDeclareCharClass\ instead.
5045 }
5046 \NewDocumentCommand \XeCJKsetcharclass { m m m }
5047 {
5048   \_XeCJK_error:n { XeCJKsetcharclass-deprecated }
5049   \XeCJK_set_char_class:nnn {#1} {#2} {#3}
5050   \XeCJKResetPunctClass
5051 }

```

5.18 兼容性修补

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。

```

5052 \cs_new_protected:Npn \XeCJK@update@fam
5053 {
5054   \addto@hook \everymath
5055   {
5056     \_XeCJK_update_main_fam:
5057     \_XeCJK_update_block_fam:
5058   }
5059 }
5060 \cs_new_protected:Npn \_XeCJK_update_main_fam:
5061 {
5062   \group_begin:
5063   \XeCJK_select_font:
5064   \exp_last_unbraced:NNNo \group_end:
5065   \tex_textfont:D \c_XeCJK_math_fam_int \tex_the:D \tex_font:D
5066 }
5067 \cs_new_protected:Npn \_XeCJK_update_block_fam:
5068 {
5069   \prop_if_empty:NF \g_XeCJK_block_fam_prop
5070   {
5071     \prop_map_function:NN
5072     \g_XeCJK_block_fam_prop
5073     \_XeCJK_update_block_fam:nn
5074   }
5075 }
5076 \cs_new_protected:Npn \_XeCJK_update_block_fam:nn #1#2
5077 {
5078   \int_set:Nn \l_XeCJK_fam_int {#2}
5079   \group_begin:
5080   \XeCJK_select_font:n {#1}
5081   \exp_last_unbraced:NNNo \group_end:
5082   \tex_textfont:D \l_XeCJK_fam_int \tex_the:D \tex_font:D
5083 }
5084 \_XeCJK_after_end_preamble:n
5085 {
5086   \bool_lazy_and:nnT
5087   { \g_XeCJK_math_bool }
5088   { \cs_if_exist_p:N \Url@MathSetup }

```

```

5089     { \tl_put_right:Nn \Url@MathSetup { \xeCJK@update@fam } }
5090   }

```

`\(` 在 $\text{\LaTeX} 2_{\epsilon}$ 中的定义是

```

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

```

`\math`

`\endmath`

`\ensuremath`

`\_xeCJK_math_robust:N`

这个定义最开始的 `\relax` 是为了防止 `\(` 出现在表格单元格的开始位置时, 模式判断不正确 (因为 $\text{\TeX}$ 会先看单元格中第一个不可展的非空格记号是否是 `\omit` 或 `\noalign`)。但是它会造成一个边界, 使 `xeCJK` 不能看到 `\relax` 后面出现的 `$`, 从而不能加入间距¹⁷。使用 $\epsilon\text{-}\text{\TeX}$ 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。

```

5091 \cs_new_protected:Npn \_xeCJK_math_robust:N #1
5092   {
5093     \group_begin: \exp_args:NcNc \group_end:
5094     { \_xeCJK_math_robust_aux:NN } #1 { \cs_to_str:N #1 ~ }
5095   }
5096 \cs_new_protected:Npn \_xeCJK_math_robust_aux:NN #1#2
5097   {
5098     \exp_args:Ne \str_case:nnTF { \cs_replacement_spec:N #1 }
5099     {
5100       { \x@protect #1 \protect #2 } { }
5101       { \protect #2 } { }
5102     }
5103     { \_xeCJK_math_robust:NN #1#2 }
5104     { \_xeCJK_math_robust:NN #1#1 }
5105   }
5106 \cs_new_protected:Npn \_xeCJK_math_robust:NN #1#2
5107   {
5108     \str_if_eq:eeTF { \cs_argument_spec:N #2 } { } { }
5109     {
5110       \exp_args:No \tl_if_head_eq_meaning:nNTF {#2} \scan_stop:
5111       {
5112         \cs_gset_protected:Npe #1
5113         { \tl_tail:N #2 }
5114       }
5115       {
5116         \cs_if_eq:NNTF #1 \ensuremath
5117         {
5118           \cs_gset_protected:Npe #1
5119           { \exp_not:o {#2} }
5120         }
5121         {
5122           \_xeCJK_warning:nec { robust-failure }
5123           { \token_to_str:N #1 } { \token_to_meaning:N #2 }
5124         }
5125       }
5126     }
5127     {
5128       \_xeCJK_warning:nec { robust-failure }
5129       { \token_to_str:N #1 } { \token_to_meaning:N #2 }
5130     }
5131   }
5132 \_xeCJK_msg_new:nnn { robust-failure }
5133 { xeCJK~can~not~make~`#1'~robust. }
5134 {
5135   The~current~meaning~of~`#1'~is:\
5136   \iow_indent:n {#2}
5137 }

```

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

```

5138 \cs_if_eq:NNTF \(\ \math
5139 {
5140   \__xeCJK_math_robust:N \(\
5141   \cs_set_eq:NN \math \(\
5142 }
5143 {
5144   \__xeCJK_math_robust:N \(\
5145   \__xeCJK_math_robust:N \math
5146 }
5147 \cs_if_eq:NNTF \) \endmath
5148 {
5149   \__xeCJK_math_robust:N \)
5150   \cs_set_eq:NN \endmath \)
5151 }
5152 {
5153   \__xeCJK_math_robust:N \)
5154   \__xeCJK_math_robust:N \endmath
5155 }
5156 \__xeCJK_math_robust:N \ensuremath

```

`\fontfamily` 使用 $\text{\LaTeX 2}_\epsilon$ 2020/10/01 提供的 NFSS 钩子, 使主要 CJK 字体族能随西文主要字体更新。
`\xeCJK@family`

```

5157 \cs_set_eq:NN \xeCJK@family \xeCJK_switch_family:e
5158 \ctex_gadd_ltxhook:nn { rmfamily } { \xeCJK@family { \CJKrmdefault } }
5159 \ctex_gadd_ltxhook:nn { sffamily } { \xeCJK@family { \CJKsfdefault } }
5160 \ctex_gadd_ltxhook:nn { ttfamily } { \xeCJK@family { \CJKttdefault } }
5161 \ctex_gadd_ltxhook:nn { normalfont } { \xeCJK@family { \CJKfamilydefault } }
5162 <@@=>

```

`\xeCJK@fix@penalty` $\text{\LaTeX 2}_\epsilon$ 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。

```

5163 \cs_new_eq:NN \xeCJK@fix@penalty \fix@penalty
5164 \tl_replace_once:Nnn \xeCJK@fix@penalty { \@@italiccorr } { \xeCJK@italiccorr }
5165 \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。

```

5166 \cs_new_protected:Npn \xeCJK@italiccorr
5167 {
5168   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
5169   { \xeCJK_italic_correction: }
5170   { \@@italiccorr }
5171 }
5172 <@@=\xeCJK>

```

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

```

5173 \cs_new_protected:Npn \xeCJK_italic_correction:
5174 { \__xeCJK_if_last_kern:T { \__xeCJK_italic_correction: } }
5175 \cs_new_protected:Npn \__xeCJK_italic_correction:
5176 {
5177   \dim_case:nnF { \tex_lastkern:D }
5178   {
5179     { \__xeCJK_node:n { default } }
5180     {
5181       \xeCJK_remove_node: \tex_italiccorrection:D
5182       \xeCJK_make_node:n { default }
5183     }
5184     { \__xeCJK_node:n { CJK } }
5185   }

```

```

5186     \xeCJK_remove_node: \tex_italiccorrection:D
5187     \xeCJK_make_node:n { CJK }
5188     \_xeCJK_italic_correction_aux:
5189   }
5190   { \_xeCJK_node:n { CJK-space } }
5191   {
5192     \xeCJK_remove_node: \tex_italiccorrection:D
5193     \xeCJK_make_node:n { CJK-space }
5194     \_xeCJK_italic_correction_aux:
5195   }
5196   }
5197   { \tex_italiccorrection:D }
5198 }

```

`\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:`)。

```

5199 \cs_new_protected:Npn \_xeCJK_italic_correction_aux:
5200 {
5201     \exp_after:wN \exp_after:wN \exp_after:wN
5202     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
5203     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
5204     \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
5205     \xeCJK_ignore_spaces:w
5206 }

```

`\g_xeCJK_xetex_allocator_int` $\text{\LaTeX}2_{\epsilon}$ 2015/01/01 接管了 `\newXeTeXintercharclass`。

```

5207 \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。

```

5208 \_xeCJK_after_end_preamble:n
5209 {
5210     \int_compare:nNnF
5211     { \c_xeCJK_class_begin_int + \seq_count:N \g_xeCJK_new_class_seq } =
5212     { \g_xeCJK_xetex_allocator_int }
5213     {
5214         \int_step_inline:nnn
5215         { \c_xeCJK_class_begin_int + 1 }
5216         { \g_xeCJK_xetex_allocator_int }
5217         {
5218             \seq_if_in:NnF \g_xeCJK_new_class_seq {#1}
5219             { \_xeCJK_set_others_toks:n {#1} }
5220         }
5221     }
5222 }
5223 \cs_new_protected:Npn \_xeCJK_set_others_toks:n #1
5224 {
5225     \int_set:cn { \_xeCJK_class_csname:n { Others } } {#1}
5226     \seq_map_inline:Nn \g_xeCJK_CJK_class_seq
5227     {
5228         \xeCJK_copy_inter_class_toks:nnnn {##1} { Others } {##1} { NormalSpace }
5229         \xeCJK_copy_inter_class_toks:nnnn { Others } {##1} { NormalSpace } {##1}
5230         \xeCJK_app_inter_class_toks:nne {##1} { Others }
5231         { \xeCJK_get_inter_class_toks:nn { Default } { Others } }
5232         \xeCJK_pre_inter_class_toks:nne { Others } {##1}
5233         { \xeCJK_get_inter_class_toks:nn { Others } { Default } }
5234         \tl_if_blank:eT
5235         { \xeCJK_get_inter_class_toks:nn { Others } { Boundary } }
5236         {
5237             \xeCJK_copy_inter_class_toks:nnnn
5238             { Others } { Boundary } { Default } { Boundary }
5239         }
5240     }
5241 }

```

```

5240         \tl_if_blank:eT
5241         { \xeCJK_get_inter_class_toks:nn { Boundary } { Others } }
5242         {
5243             \xeCJK_copy_inter_class_toks:nnnn
5244             { Boundary } { Others } { Boundary } { Default }
5245         }
5246     }
5247 }

```

用于保护下面歧义宽度标点的分组。

```

\__xeCJK_inactive_group_begin:
\__xeCJK_inactive_group_end:
5248 \cs_new_protected:Npn \__xeCJK_inactive_group_begin:
5249 { \group_begin: \makexeCJKinactive }
5250 \cs_new_eq:NN \__xeCJK_inactive_group_end: \group_end:

```

单独处理宽度有分歧的几个标点：包括省略号、破折号、间隔号、引号等中西文混用的符号，保证其命令形式输出的是西文字体。如果 xunicode 宏包被载入，则通过 xunicode-addon 处理。

```

5251 \prop_const_from_keyval:Nn \c__xeCJK_ambiguous_char_prop
5252 {
5253     "00B7 = \textperiodcentered \textcentereddot \textcdot ,
5254     "2013 = \textendash ,
5255     "2014 = \textemdash ,
5256     "2018 = \textquoteleft \textgrq ,
5257     "2019 = \textquoteright ,
5258     "201C = \textquotedblleft \textgrqq ,
5259     "201D = \textquotedblright ,
5260     "2025 = \texthdottedfor ,
5261     "2026 = \textellipsis ,
5262     "2027 = \textthyphenationpoint ,
5263     "2E3A = \texttwoemdash
5264 }
5265 \__xeCJK_at_end_preamble:n { \__xeCJK_patch_text_command: }
5266 \cs_new_protected:Npn \__xeCJK_patch_text_command:
5267 {
5268     \xeCJK_if_package_loaded:nTF { xunicode }
5269     { \__xeCJK_patch_xunicode_ambiguous_char: }
5270     {
5271         \exp_args:Ne \__xeCJK_patch_tuenc_ambiguous_char:n
5272         { \UnicodeEncodingName }
5273         \__xeCJK_patch_tuenc_accent:
5274         \__xeCJK_patch_tuenc_composite:
5275     }
5276 }
5277 \cs_new_protected:Npn \__xeCJK_patch_xunicode_ambiguous_char:
5278 {
5279     \RequirePackage { xunicode-addon }
5280     \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
5281     {
5282         \tl_map_inline:nn { ##2 }
5283         {
5284             \xunadd_set_begin_hook:nn { ####1 }
5285             { \__xeCJK_inactive_group_begin: }
5286             \xunadd_set_end_hook:nn { ####1 }
5287             { \__xeCJK_inactive_group_end: }
5288         }
5289     }
5290     \xunadd_append_begin_hook:n { \xeCJK_make_boundary: }
5291 }
5292 \cs_new_protected:Npn \__xeCJK_patch_tuenc_ambiguous_char:n #1
5293 {
5294     \prop_map_inline:Nn \c__xeCJK_ambiguous_char_prop
5295     {
5296         \tl_map_inline:nn { ##2 }
5297         {
5298             \cs_if_exist:NF ####1
5299             { \DeclareTextSymbol ####1 {#1} { ##1 } }
5300             \__xeCJK_patch_ambiguous_char:nN {#1} ####1

```

```

5301     }
5302   }
5303 }
5304 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nN #1#2
5305 {
5306   \exp_args:Ne \__xeCJK_patch_ambiguous_char:nn
5307   { #1 \token_to_str:N #2 }
5308   { #1 - #2 }
5309 }
5310 \cs_new_protected:Npe \__xeCJK_patch_ambiguous_char:nNn #1#2#3
5311 {
5312   \exp_not:N \exp_args:Ne
5313   \exp_not:N \__xeCJK_patch_ambiguous_char:nn
5314   {
5315     \c_backslash_str #1
5316     \exp_not:N \token_to_str:N #2 -
5317     \exp_not:N \token_to_str:N #3
5318   }
5319   { #1 - #2#3 }
5320 }
5321 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:nn #1#2
5322 {
5323   \cs_if_free:cF {#1}
5324   { \exp_args:Nc \__xeCJK_patch_ambiguous_char:Nn {#1} {#2} }
5325 }
5326 \cs_new_protected:Npn \__xeCJK_patch_ambiguous_char:Nn #1#2
5327 {
5328   \token_if_chardef:NTF #1
5329   {
5330     \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
5331     { \int_eval:n {#1} }
5332     \cs_set_protected:Npe #1
5333     { \__xeCJK_ambiguous_char:n { \tex_Uchar:D #1 } }
5334   }
5335   {
5336     \prop_gput:Nne \g__xeCJK_ambiguous_slot_prop {#2}
5337     { \int_eval:n { \exp_after:wN ` #1 } }
5338     \cs_set_protected:Npe #1
5339     { \__xeCJK_ambiguous_char:n { \exp_not:o {#1} } }
5340   }
5341 }
5342 \cs_new_protected:Npn \__xeCJK_ambiguous_char:n #1
5343 {
5344   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
5345   { \__xeCJK_inactive_group_begin: #1 \__xeCJK_inactive_group_end: }
5346   {#1}
5347 }
5348 \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

```

5349 \cs_new_protected:Npn \xeCJK_text_composite_patch:
5350 {
5351   \str_if_eq:eeT { \f@encoding } { \UnicodeEncodingName }
5352   { \xeCJK_make_boundary: }
5353 }

```

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。

```

5354 \cs_new_protected:Npn \__xeCJK_patch_tuenc_composite:
5355 {
5356   \cs_set_nopar:Npn \@text@composite@x
5357   {
5358     \xeCJK_text_composite_patch:
5359     \cs_if_exist_use:NF

```

```

5360     }
5361 }

```

`\_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。

```

5362 \group_begin:
5363 \char_set_catcode_other:n { "A0 }
5364 \cs_new_protected:Npn \_xeCJK_patch_tuenc_accent:
5365 {
5366     \cs_set_protected_nopar:Npn \add@unicode@accent ##1 ##2
5367     {
5368         \tl_if_blank:nTF { ##2 } { ^~a0 } { ##2 }
5369         \tex_Uchar:D \tex_numexpr:D ##1 \scan_stop:
5370     }
5371 }
5372 \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。

```

5373 \prop_const_from_keyval:Nn \c__xeCJK_middle_dot_prop
5374 {
5375     T2A = \cyrchvcrs ,
5376     T2B = \cyrchldsc ,
5377     T2C = \cyrabhha ,
5378     X2  = \cyrchldsc ,
5379     TS1 = \textperiodcentered ,
5380     LY1 = \textperiodcentered ,
5381     T1  = \r u ,
5382     T4  = \B t ,
5383     T5  = \` \ecircumflex
5384 }
5385 \_xeCJK_at_end_preamble:n { \_xeCJK_patch_middle_dot: }
5386 \cs_new_protected:Npn \_xeCJK_patch_middle_dot:
5387 {
5388     \prop_map_inline:Nn \c__xeCJK_middle_dot_prop
5389     { \_xeCJK_patch_middle_dot:nw { ##1 } ##2 \q_stop }
5390     \_xeCJK_patch_ambiguous_char:nNn { T5 } \` { \^ - e }
5391 }
5392 \cs_new_protected:Npn \_xeCJK_patch_middle_dot:nw #1#2#3 \q_stop
5393 {
5394     \tl_if_empty:nTF {#3}
5395     { \_xeCJK_patch_ambiguous_char:nN {#1} #2 }
5396     { \_xeCJK_patch_ambiguous_char:nNn {#1} #2 {#3} }
5397 }

```

`pifont` 宏包的符号 `\ding{183}` 也有冲突。

```

5398 \_xeCJK_package_hook:nn { pifont }
5399 {
5400     \RenewDocumentCommand \Pifont { m }
5401     { \mode_leave_vertical: \makexeCJKinactive \usefont { U } {#1} { m } { n } }
5402 }

```

`\xeCJKchar` `\char` 是 $\TeX$ 原语, 用户使用它直接通过字符编码从当前字体取字形, 不经过 NFSS 层。一些宏包(如 `mtpro2`)在数学模式中用 `\char` 取非 CJK 字形时, 会被 `xeCJK` 的 `interchar` 机制拦截并切换到中文字体, 导致字形变成中文标点(如间隔号, #407)。

参照 `Lua $\TeX$ -ja` 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 `Lua $\TeX$ -ja` 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。

```

5403 \cs_new_protected:Npn \xeCJK_default_char:w
5404 {
5405   \int_compare:nNnTF \tex_XeTeXinterchartokenstate:D > \c_zero_int
5406   {
5407     \c_group_begin_token
5408     \tex_afterassignment:D \__xeCJK_default_char_aux:
5409     \l__xeCJK_default_char_int =
5410   }
5411   { \tex_char:D }
5412 }
5413 \cs_new_protected:Npn \__xeCJK_default_char_aux:
5414 {
5415   \tex_XeTeXinterchartokenstate:D = \c_zero_int
5416   \tex_char:D \l__xeCJK_default_char_int
5417   \c_group_end_token
5418 }
5419 \int_new:N \l__xeCJK_default_char_int
5420 \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。

```

5421 \__xeCJK_package_hook:nn { mtpo2 }
5422 {
5423   \cs_set_eq:NN \__xeCJK_mtpo_overbrace:w \overbrace
5424   \cs_gset_protected:Npn \overbrace #1
5425   {
5426     \makexeCJKinactive
5427     \__xeCJK_mtpo_overbrace:w {#1}
5428   }
5429   \cs_set_eq:NN \__xeCJK_mtpo_underbrace:w \underbrace
5430   \cs_gset_protected:Npn \underbrace #1
5431   {
5432     \makexeCJKinactive
5433     \__xeCJK_mtpo_underbrace:w {#1}
5434   }
5435 }

```

`\__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:
5436 \__xeCJK_package_hook:nn { unicode-math }
5437 {
5438   \prop_const_from_keyval:Nn \c__xeCJK_um_ambiguous_char_prop
5439   {
5440     "00B7 = \cdotp ,
5441     "2025 = \enleadertwodots ,
5442     "2026 = \unicodeellipsis
5443   }
5444   \cs_new_protected:Npn \__xeCJK_save_um_char:
5445   {

```

```

5446 \cs_set_protected:Npe \__xeCJK_restore_um_char:
5447 {
5448   \prop_map_function:NN
5449   \c__xeCJK_um_ambiguous_char_prop
5450   \__xeCJK_restore_um_char_aux:nn
5451 }
5452 }
5453 \cs_new_eq:NN \__xeCJK_restore_um_char: \prg_do_nothing:
5454 \cs_new:Npn \__xeCJK_restore_um_char_aux:nn #1#2
5455 {
5456   \__xeCJK_gset_mathcodenum:nn
5457   { \int_value:w #1 }
5458   { \int_value:w \tex_Umathcodenum:D #1 }
5459 }
5460 \cs_new_protected:Npn \__xeCJK_gset_mathcodenum:nn #1#2
5461 {
5462   \int_compare:nNnF { \tex_Umathcodenum:D #1 } = {#2}
5463   { \tex_global:D \tex_Umathcodenum:D #1 = #2 ~ }
5464 }
5465 }

```

__xeCJK_patch_microtype_get_slot: 兼容 microtype。

```

5466 \cs_new_protected:Npn \__xeCJK_patch_microtype_get_slot:
5467 {
5468   \cs_new_eq:NN \xeCJK@original@get@slot \MT@get@slot@
5469   \cs_set_eq:NN \MT@get@slot@ \xeCJK@microtype@get@slot
5470   \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
5471 }
5472 \cs_new_protected_nopar:Npn \xeCJK@microtype@get@slot
5473 {
5474   \int_compare:nNnT \MT@char < \c_zero_int
5475   { \__xeCJK_get_ambiguous_slot: }
5476   \xeCJK@original@get@slot
5477 }
5478 \cs_new_protected:Npn \__xeCJK_get_ambiguous_slot:
5479 {
5480   \prop_get:NeNT \g__xeCJK_ambiguous_slot_prop
5481   { \MT@encoding - \tex_the:D \MT@toks } \l__xeCJK_tmp_tl
5482   { \cs_set_eq:NN \MT@char \l__xeCJK_tmp_tl }
5483 }
5484 \cs_new_protected:Npn \xeCJK@microtype@restore@pickupfont
5485 { \__xeCJK_gadd_font_initial_hook:n { \MT@ltx@pickupfont } }
5486 \__xeCJK_package_hook:nn { microtype }
5487 {
5488   \cs_if_free:NF \MT@get@slot@
5489   { \__xeCJK_patch_microtype_get_slot: }
5490   \MT@addto@setup { \xeCJK@microtype@restore@pickupfont }
5491 }

```

简单处理与 hyperref 宏包的兼容问题。

\Hy@BeginAnnot 的补丁处理链接起始端: 保存真实的 CJK 状态、清除历史状态, 并且不重放可能陈旧的 default。链接结束端需要处理一个不同的问题。url 的 \Url@FormatString 以数学模式排版 URL; 单独加载 url 时, 后续 CJK 字符可以直接看到数学节点并恢复 ecglue。hyperref 随后由 \Hy@EndAnnot 插入链接结束 whatsit, 数学节点因此不再是最后节点。

在顶层 annotation 结束前若最后节点确为数学节点, 说明链接的可见内容提供了真实的西文边界。原始 \Hy@EndAnnot 插入结束 whatsit 后, 重放一个专用的 hyperref-default 标记, 让后续 CJK 转换走既有边界恢复链; 若后续紧接颜色切换或另一个 annotation, 现有补丁也能继续保存并重放这个可信标记。它与普通 default 分离, 避免重新放宽 #810 为防御陈旧 default 而设的选择性重放规则。只处理 \c@Hy@AnnotLevel=1, 避免嵌套 annotation 的内部结束点提前改写外层状态; 非数学内容保持现有 CJK/default 状态机不变。

```

5492 \tl_new:N \g__xeCJK_hyper_saved_node_tl

```

```

5493 \cs_new_protected:Npn \__xeCJK_hyperref_end_annot_math:
5494 {
5495   \__xeCJK_orig_Hy_EndAnnot:
5496   \xeCJK_make_node:n { hyperref-default }
5497 }
5498 \cs_new_protected:Npn \__xeCJK_patch_hyperref_annot:
5499 {
5500   \cs_gset_eq:NN \__xeCJK_orig_Hy_BeginAnnot:n \Hy@BeginAnnot
5501   \cs_gset_eq:NN \__xeCJK_orig_Hy_EndAnnot: \Hy@endAnnot
5502   \cs_gset_protected:Npn \Hy@BeginAnnot ##1
5503   {
5504     \xeCJK_if_last_node:TF
5505     { \tl_gset_eq:NN \g__xeCJK_hyper_saved_node_tl \g__xeCJK_last_node_tl }
5506     { \tl_gclear:N \g__xeCJK_hyper_saved_node_tl }
5507     \tl_gclear:N \g__xeCJK_last_node_tl
5508     \__xeCJK_orig_Hy_BeginAnnot:n { ##1 }
5509     \tl_if_eq:NnT \g__xeCJK_hyper_saved_node_tl { CJK }
5510     { \xeCJK_make_node:n { CJK } }
5511     \tl_if_eq:NnT \g__xeCJK_hyper_saved_node_tl { CJK-space }
5512     { \xeCJK_make_node:n { CJK-space } }
5513     \tl_if_eq:NnT \g__xeCJK_hyper_saved_node_tl { CJK-widow }
5514     { \xeCJK_make_node:n { CJK-widow } }
5515     \tl_if_eq:NnT \g__xeCJK_hyper_saved_node_tl { hyperref-default }
5516     { \xeCJK_make_node:n { hyperref-default } }
5517   }
5518   \cs_gset_protected:Npn \Hy@endAnnot
5519   {
5520     \int_compare:nNnTF { \c@Hy@AnnotLevel } = { 1 }
5521     {
5522       \__xeCJK_if_last_math:TF
5523       { \__xeCJK_hyperref_end_annot_math: }
5524       { \__xeCJK_orig_Hy_EndAnnot: }
5525     }
5526     { \__xeCJK_orig_Hy_EndAnnot: }
5527   }
5528 }
5529 \__xeCJK_package_hook:nn { hyperref }
5530 {
5531   \pdfstringdefDisableCommands
5532   {
5533     \__xeCJK_gobble_CJKfamily:
5534     \xeCJK_cs_clear:N \__xeCJK_inactive_group_begin:
5535     \xeCJK_cs_clear:N \__xeCJK_inactive_group_end:
5536     \xeCJK_cs_clear:N \makexeCJKinactive
5537     \xeCJK_cs_clear:N \xeCJK_text_composite_patch:
5538   }
5539   \__xeCJK_patch_hyperref_annot:
5540 }

```

color/xcolor 的颜色切换通过 \set@color 插入 \special 节点(whatsit), 会遮蔽 xeCJK 的标记 kern, 导致后续的 CJKecglue 检测失败。此处可在 \set@color 和 \reset@color 之后重新放置 xeCJK 节点标记。 \set@color 负责颜色推入(push), \reset@color 负责颜色弹出(pop)——后者通过 \aftergroup 在分组结束时自动调用, 用于恢复原始颜色。当最后一个节点是 xeCJK 标记 kern 对时直接重放标记; 当最后一个节点是 hlist 且 \g_@@_last_node_tl 非空时, 仅设置 \g_@@_reset_color_pending_bool 以延迟到 \@_check_for_glue_skip: 的 whatsit 回退路径中处理, 避免在页面布局结构中插入多余 kern。

```

5541 \cs_new_protected:Npn \__xeCJK_patch_set_color:
5542 {
5543   \cs_if_exist:NF \__xeCJK_orig_set_color:
5544   {
5545     \cs_set_eq:NN \__xeCJK_orig_set_color: \set@color
5546     \cs_gset_protected:Npn \set@color
5547     {
5548       \xeCJK_if_last_node:TF

```

```

5549         {
5550             \__xeCJK_orig_set_color:
5551             \tl_if_empty:NF \g__xeCJK_last_node_tl
5552             { \exp_args:No \xeCJK_make_node:n { \g__xeCJK_last_node_tl } }
5553         }
5554         { \tl_gclear:N \g__xeCJK_last_node_tl \__xeCJK_orig_set_color: }
5555     }
5556 }
5557 \cs_if_exist:NF \__xeCJK_orig_reset_color:
5558 {
5559     \cs_set_eq:NN \__xeCJK_orig_reset_color: \reset@color
5560     \cs_gset_protected:Npn \reset@color
5561     {
5562         \xeCJK_if_last_node:TF
5563         {
5564             \__xeCJK_orig_reset_color:
5565             \tl_if_empty:NF \g__xeCJK_last_node_tl
5566             {
5567                 \exp_args:No \xeCJK_make_node:n { \g__xeCJK_last_node_tl }
5568                 \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
5569             }
5570         }
5571         {
5572             \__xeCJK_if_last_hlist:TF
5573             {
5574                 \tl_if_empty:NF \g__xeCJK_last_node_tl
5575                 { \bool_gset_true:N \g__xeCJK_reset_color_pending_bool }
5576                 \__xeCJK_orig_reset_color:
5577             }
5578             { \__xeCJK_orig_reset_color: }
5579         }
5580     }
5581 }
5582 }
5583 \__xeCJK_package_hook:nn { color } { \__xeCJK_patch_set_color: }
5584 \__xeCJK_package_hook:nn { xcolor } { \__xeCJK_patch_set_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。此处对这两个函数施加与上面相同的 `kern` 对保护。

```

5585 \cs_new_protected:Npn \__xeCJK_patch_l3color:
5586 {
5587     \cs_if_exist:NF \__xeCJK_orig_color_select:N
5588     {
5589         \cs_set_eq:NN \__xeCJK_orig_color_select:N \__color_select:N
5590         \cs_gset_protected:Npn \__color_select:N ##1
5591         {
5592             \xeCJK_if_last_node:TF
5593             {
5594                 \__xeCJK_orig_color_select:N ##1
5595                 \tl_if_empty:NF \g__xeCJK_last_node_tl
5596                 { \exp_args:No \xeCJK_make_node:n { \g__xeCJK_last_node_tl } }
5597             }
5598             { \tl_gclear:N \g__xeCJK_last_node_tl \__xeCJK_orig_color_select:N ##1 }
5599         }
5600     }
5601     \cs_if_exist:NF \__xeCJK_orig_color_backend_reset:
5602     {
5603         \cs_set_eq:NN \__xeCJK_orig_color_backend_reset: \__color_backend_reset:
5604         \cs_gset_protected:Npn \__color_backend_reset:
5605         {
5606             \xeCJK_if_last_node:TF
5607             {
5608                 \__xeCJK_orig_color_backend_reset:
5609                 \tl_if_empty:NF \g__xeCJK_last_node_tl
5610                 {

```

```

5611         \exp_args:No \xeCJK_make_node:n { \g__xeCJK_last_node_tl }
5612         \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
5613     }
5614 }
5615 {
5616     \__xeCJK_if_last_hlist:TF
5617     {
5618         \tl_if_empty:NF \g__xeCJK_last_node_tl
5619         { \bool_gset_true:N \g__xeCJK_reset_color_pending_bool }
5620         \__xeCJK_orig_color_backend_reset:
5621     }
5622     { \__xeCJK_orig_color_backend_reset: }
5623 }
5624 }
5625 }
5626 }
5627 \__xeCJK_patch_l3color:

```

hypdoc 的 \HD@target 通过 \raisebox 产生 hbox(0+0)x0 用于放置 \special{pdf:dest...}。该 hbox 隔在 xeCJK 标记 kern 对和后续字符之间,使 \tex_lastkern:D 探测失败,导致 Boundary→{Default,CJK} 路径中 \xeCJK_if_last_node:n / \@@_if_last_kern: 检查返回 false,间距因此丢失。此处保存 \g__@@_last_node_tl、清除现有标记 kern、调用原始 \HD@target、然后根据保存的状态重放节点标记,使后续触发路径能正确识别前方上下文。

```

5628 \tl_new:N \g__xeCJK_hd_saved_node_tl
5629 \cs_new_protected:Npn \__xeCJK_patch_hd_target:
5630 {
5631     \cs_if_exist:NF \__xeCJK_orig_HD_target:
5632     {
5633         \cs_set_eq:NN \__xeCJK_orig_HD_target: \HD@target
5634         \cs_gset_protected:Npn \HD@target
5635         {
5636             \xeCJK_if_last_node:TF
5637             { \tl_gset_eq:NN \g__xeCJK_hd_saved_node_tl \g__xeCJK_last_node_tl }
5638             { \tl_gclear:N \g__xeCJK_hd_saved_node_tl }
5639             \tl_gclear:N \g__xeCJK_last_node_tl
5640             \__xeCJK_orig_HD_target:
5641             \tl_if_empty:NF \g__xeCJK_hd_saved_node_tl
5642             { \exp_args:No \xeCJK_make_node:n { \g__xeCJK_hd_saved_node_tl } }
5643         }
5644     }
5645 }
5646 \__xeCJK_package_hook:nn { hypdoc } { \__xeCJK_patch_hd_target: }

```

url 的 \url 命令通过 \Url@FormatString 进入数学模式排版 URL 字符串。数学模式 node 隔在 xeCJK 标记 kern 对和数学公式之间,又因 $\text{\XeTeX}$ 在数学模式中不触发类别转换 (interchar class), 标记 kern 对永远等不到匹配的 Boundary→{CJK,Default} 触发,缓存的 CJKecglue 因此在 CJK 与 \url 之间丢失。此处在进入数学模式前先 drain(排空)缓存间距: \url 排版的内容必然是西文,前方上下文为 CJK / CJK-space / CJK-widow / default 中任意一种时均应补上 CJKecglue(西文边界间距)。\url 数学模式结束后后续 CJK 文字之间的右侧间距,由 \xeCJK_check_for_glue: 的 __xeCJK_if_last_math: 路径自然处理。

```

5647 \cs_new_protected:Npn \__xeCJK_drain_ecglue:
5648 {
5649     \xeCJK_if_last_node:TF
5650     {
5651         \tl_if_empty:NF \g__xeCJK_last_node_tl
5652         {
5653             \xeCJK_remove_node:
5654             \skip_horizontal:N \l__xeCJK_ecglue_skip
5655         }
5656     }
5657     % 无 marker kern: tl 与实际节点状态不一致,主动清空避免后续误读残留。
5658     { \tl_gclear:N \g__xeCJK_last_node_tl }

```

```

5659 }
5660 \cs_new_protected:Npn \__xeCJK_patch_url_format:
5661 {
5662   \cs_if_exist:NF \__xeCJK_orig_Url_FormatString:
5663   {
5664     \cs_set_eq:NN \__xeCJK_orig_Url_FormatString: \Url@FormatString
5665     \cs_gset_protected:Npn \Url@FormatString
5666     {
5667       \__xeCJK_drain_ecglue:
5668       \__xeCJK_orig_Url_FormatString:
5669     }
5670   }
5671 }
5672 \__xeCJK_package_hook:nn { url } { \__xeCJK_patch_url_format: }

```

$\text{\LaTeX}2_{\epsilon}$ 的 `\verb` 命令展开后会执行 `\leavevmode`, 其中 `\null` 等价于 `\hbox{}`, 会产生一个 `\hbox(0+0)x0` 节点。这个 0×0 的 `\hbox` 紧跟在 `xeCJK` 的标记 `kern` 对之后, 会切断后续 `Boundary \rightarrow \{Default, CJK\}` 类别转换中 `\tex_lastkern:D` 的探测路径, 导致 `CJK` 文字与 `\verb` 之间应有的 `CJKecglue` 丢失。

修复办法是在执行原始 `\verb` 之前先 `drain` 缓存的间距: 因为 `\verb` 排版的内容必然是西文 (catcode 12 字符), 前文若为 `CJK`, 那么这里就应该补上 `CJKecglue`。而 `\verb` 内最后一个字符到后续 `CJK` 文字之间的右侧间距, 会由 `Default \rightarrow CJK` 类别转换中 `\xeCJK_pre_inter_class_toks:nnn` 自动恢复, 不需要额外处理。

数学模式下, 原 `\verb` 走 `\hbox` 分支, 不会展开 `\leavevmode`, 标记 `kern` 不被 0×0 的 `\hbox` 遮蔽, 因此跳过这个 patch。

`shortverb` 的 `\MakeShortVerb` 把分隔符设为 active 字符后展开到 `\verb`, 会自动跟随这里的 patch。`\verb*` 与 `\verb` 共用入口 (`\@ifstar` 在 `\bgroup` 之后才扫描), 因此也自动覆盖到。

这里选择 `\__xeCJK_after_preamble:n` 而非 `\__xeCJK_package_hook:nn`, 因为 `\verb` 是 $\text{\LaTeX}2_{\epsilon}$ 核心命令, 不来自宏包; 放在 `preamble` 之后注册, 可以避开 `cprotect`、`fancyverb` 等宏包对 `\verb` 重定义时的时序冲突。

这里不直接复用 `\__xeCJK_drain_ecglue:(#880 中 \Url@FormatString 使用的 drain 函数)`: 后者在没有标记时会主动清掉 `\g__xeCJK_last_node_tl`, 这是为 `math` 模式吃掉 `marker` 的场景做的防御。而 `\verb` 不会进入 `math` 模式, 入口前的 `\g__xeCJK_last_node_tl` 是合法状态——在 `ctex` 模式下, `\verb` 内部会切换 `CJK` 字体 (如 `FandolFang`), `\verb` 内字符到 `\verb` 外字符仍会走 `CJK \rightarrow CJK` 类别转换路径, 此时需要保留 `\g__xeCJK_last_node_tl` 的当前值, 才能正确地输出 `\CJKglue`。因此 `\verb` 专用的 `drain` 只在确实存在标记时移除并补 `ecglue`, 没有标记时 `else` 分支故意留空——与 `\__xeCJK_drain_ecglue:` 的唯一区别就在于此, 不调用 `\tl_gclear:N` 是为了不破坏入口前的合法状态。

此 `drain` 函数同样被 `\__xeCJK_patch_codedoc_meta:(#920 中 l3doc \meta 补丁)` 复用: `\meta` 退出后仍然有 token 层面的 `interchar` 处理, 需要保留 `\g__xeCJK_last_node_tl` 的状态。后续若有新的调用点, 请同步更新本节注释。

```

\__xeCJK_drain_ecglue_verb: 5673 \cs_new_protected:Npn \__xeCJK_drain_ecglue_verb:
5674 {
5675   \xeCJK_if_last_node:TF
5676   {
5677     \tl_if_empty:NF \g__xeCJK_last_node_tl
5678     {
5679       \xeCJK_remove_node:
5680       \skip_horizontal:N \l__xeCJK_ecglue_skip
5681     }
5682   }
5683   { }
5684 }

```

`\verb` 右侧还有一处独立问题(#919):`\verb` 组内 `\language` 被切到 `\l@nohyphenation`, 组关闭后 `TEX` 的 `language whatsit` 是延迟排出的——要等下一个字符节点进入水平列表时才补插。于是 `\verb` 与后文之间的源码空格产生的 `inter-word glue` 先进列表, `\setlanguage whatsit` 反而排在 `glue` 之后:

`kern pair → glue(source space) → setlanguage → [探测点]`

后续 `CJK` 字符触发类别转换时, `\xeCJK_check_for_glue`: 看到的 `last node` 是 `whatsit` 而非 `glue`, 走 `\_xeCJK_recover_glue_whatsit`: 的 `default` 分支再补一个 `CJKecglue`——与已有的 `inter-word glue` 相加成双倍间距。`\lastskip` 无法穿透 `whatsit`, `default` 分支也不能收窄(无源码空格的字`\verb...` 字 场景依赖它补合法 `ecglue`), 因此不从探测端修, 而是从节点顺序入手: 在 `\verb@egroup` 关组之后立即执行 `\setlanguage\language`, 强制 `language whatsit` 当场排出。此后源码空格的 `glue` 排在 `whatsit` 之后:

`kern pair → setlanguage → glue(source space) → [探测点]`

探测点看到的 `last node` 是 `glue`, 走 `\_xeCJK_check_for_glue_skip`: 正确处理, 不再重复补 `ecglue`; 无源码空格时 `last node` 是我们排出的 `whatsit`, `default` 分支照常补上合法 `ecglue`, 原有行为不变。

`\_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_patch_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。

```
5685 \cs_new_protected:Npn \_xeCJK_flush_language_whatsit:
5686   { \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 `drain` 与 `flush` 两个补丁都自动覆盖到。

```
\_xeCJK_patch_verb: 5687 \cs_new_protected:Npn \_xeCJK_patch_verb:
5688   {
5689     \cs_if_exist:NF \_xeCJK_orig_verb:
5690     {
5691       \cs_set_eq:NN \_xeCJK_orig_verb: \verb
5692       \cs_gset_protected:Npn \verb
5693       {
5694         \mode_if_math:F { \_xeCJK_drain_ecglue_verb: }
5695         \_xeCJK_orig_verb:
5696       }
5697       \tl_gput_right:Nn \verb@egroup { \_xeCJK_flush_language_whatsit: }
5698     }
5699   }
5700 \_xeCJK_after_preamble:n { \_xeCJK_patch_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`。

但仅用 `hbox` 包装会带来另一个副作用: `\meta` 之前的 `CJK` 与 `\meta` 之间的边界 `ecglue` 会丢失——`\_codedoc_meta_original:n` 入口的 `\ensuremath math` 节点把缓存的 `CJK` 标

记 kern 遮在自己之后, 后续类别转换探测不到这个标记, 原本应有的 CJKecglue 也跟着丢失。与 url(#880)、\verb(#910, 参见 _xeCJK_patch_verb:) 同型, 这里在 _codedoc_meta:n 入口调用 _xeCJK_drain_ecglue_verb: 移除标记并补上 \CJKecglue, 恢复左侧的边界 ecglue。复用 \verb 版本的 drain (else 分支不清 \g__xeCJK_last_node_tl) 是因为 \meta 退出后仍然要经过 token 层面的类别转换, 需要保留 \g__xeCJK_last_node_tl 的当前状态。_xeCJK_drain_ecglue_verb: 的注释也列出了本处调用点, 二者保持双向引用。

\rangle 之后到外部字符的 ecglue 仍由 _codedoc_meta:n 末尾 \ensuremath 退出后的 Default→CJK 类别转换正常处理。

_codedoc_meta:n 同时被 \Arg、\oarg、\parg 等命令共用, 此补丁也间接修复后者的相同问题。

这里选择 _xeCJK_at_end_preamble:n 而非 _xeCJK_package_hook:nn, 原因在于 l3doc 是文档类, 文档类加载早于 xeCJK, 需要等到 preamble 结束时才能判定 _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_patch_codedoc_meta: 5701 \cs_new_protected:Npn \_xeCJK_patch_codedoc_meta:
5702 {
5703   \cs_if_exist:NTF \_codedoc_meta:n
5704   {
5705     \cs_if_exist:NF \_xeCJK_orig_codedoc_meta:n
5706     {
5707       \cs_set_eq:NN \_xeCJK_orig_codedoc_meta:n \_codedoc_meta:n
5708       \cs_gset_protected:Npn \_codedoc_meta:n ##1
5709       {
5710         \mode_if_math:F { \_xeCJK_drain_ecglue_verb: }
5711         \_xeCJK_orig_codedoc_meta:n { \hbox:n {##1} }
5712       }
5713     }
5714   }
5715   {
5716     \bool_lazy_and:nnT
5717     { \cs_if_exist_p:c { ver@doc.sty } }
5718     { \cs_if_free_p:N \_xeCJK_orig_codedoc_meta:n }
5719     {
5720       \cs_set_eq:NN \_xeCJK_orig_codedoc_meta:n \meta
5721       \DeclareDocumentCommand \meta { m }
5722       {
5723         \mode_if_math:F { \_xeCJK_drain_ecglue_verb: }
5724         \_xeCJK_orig_codedoc_meta:n { \hbox:n {##1} }
5725       }
5726     }
5727   }
5728 }
5729 \_xeCJK_at_end_preamble:n { \_xeCJK_patch_codedoc_meta: }
```

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 节点紧接在前一条目结尾的 default 标记 kern 之后, 遮蔽了 \tex_lastkern:D 探测路径。当下一条目首字符是 CJK 字符时, Boundary→CJK 类别转换的 \xeCJK_check_for_glue: 会走到 _xeCJK_recover_glue_whatst: 的 default 分支——它读到全局 \g__xeCJK_last_node_tl 仍是 default (前一条目的合法残留), 就误吐了一个 CJKecglue, 表现为条目首字符前多出一段空白。

修复策略与 hyperref \Hy@BeginAnnot (#809/#810) 同型: 在 whatsit 插入前主动清空 \g__xeCJK_last_node_tl, 让下游 _xeCJK_recover_glue_whatst: 探到 whatsit 时 default 分支条件不成立, 不再吐出多余 ecglue。

前一条目结尾如果确实需要往下一条目续接 CJK 边界语义 (例如后续 \blx@driver 内部先输出 CJK 字符), \blx@driver 排版首字符时会由 \xeCJK_make_node:n 重新设置 \g__xeCJK_last_node_tl, 不受此清空影响; 因此本补丁只切断“前一条目 marker 通过 whatsit 遮蔽错误跨到下一条目”的路径, 不影响新条目内部的正常边界恢复。

与 \Hy@BeginAnnot 补丁的差异: 这里没有“选择性重放 CJK/CJK-space 标记”的需求, 因为 \blx@pagetracker 入口时前文已经是完整段落结构(\bibsentence、\blx@postpunct 等), 跨条目续接 xeCJK marker 的语义不成立, 直接清空即可。

补丁点选 \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_patch_biblatex_pagetracker: 5730 \cs_new_protected:Npn \_xeCJK_patch_biblatex_pagetracker:
5731 {
5732   \cs_if_exist:NT \blx@pagetracker
5733   {
5734     \cs_if_exist:NF \_xeCJK_orig_blx_pagetracker:
5735     {
5736       \cs_set_eq:NN \_xeCJK_orig_blx_pagetracker: \blx@pagetracker
5737       \cs_gset_protected:Npn \blx@pagetracker
5738       {
5739         \tl_gclear:N \g__xeCJK_last_node_tl
5740         \_xeCJK_orig_blx_pagetracker:
5741       }
5742     }
5743   }
5744 }
5745 \_xeCJK_at_end_preamble:n { \_xeCJK_patch_biblatex_pagetracker: }
```

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

```
5746 \_xeCJK_package_hook:nn { cprotect }
5747 {
5748   \cs_if_free:NF \icprotect
5749   { \exp_after:wN \tex_let:D \cs:w cprotect \cs_end: \icprotect }
5750 }
```

在 listings 宏包后自动载入 xeCJK-listings。

```
5751 \_xeCJK_package_hook:nn { listings }
5752 { \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 everyisel 包判断错误。需要在它们判断之前取消定义。

```
5753 \_xeCJK_package_hook:nn { everyisel }
5754 {
5755   \cs_if_exist:NF \@EverySelectfont@Legacy
5756   { \cs_undefine:c { ver@CJK . \c__xeCJK_package_ext_tl } }
5757 }
```

\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。

```
5758 \ctex_at_begin_package:nn { CJKnumb }
5759 {
5760   \tl_new:N \l__xeCJK_CJK_version_tl
5761   \tl_set_eq:Nc \l__xeCJK_CJK_version_tl { ver@CJK . \c__xeCJK_package_ext_tl }
5762   \tl_set:cn { ver@CJK . \c__xeCJK_package_ext_tl } { 9999/99/99 }
5763   \cs_new_protected:Npn \CJKaddEncHook #1#2
5764   {
5765     \str_if_eq:nnT {#1} { \CJK@UnicodeEnc }
5766     {
5767       \group_begin:
```

```

5768         \cs_set_eq:NN \Unicode \xeCJK_unicode_char:nn
5769         \cs_set_eq:NN \def \xdef
5770         #2
5771         \group_end:
5772         \str_gset:Nn \CJK@tenthousand { ^^^~4e07 }
5773         \str_gset:Nn \CJK@hundredmillion { ^^^~4ebf }
5774         \tl_if_exist:NF \CJK@UnicodeEnc
5775         { \tl_const:Nn \CJK@UnicodeEnc { UTF8 } }
5776         \cs_if_exist:NF \Unicode
5777         { \cs_new_eq:NN \Unicode \xeCJK_unicode_char:nn }
5778     }
5779 }
5780 \cs_new:Npn \xeCJK_unicode_char:nn #1#2
5781 { \tex_Uchar:D \tex_numexpr:D (#1) * 256 + (#2) \scan_stop: }
5782 }
5783 \ctex_at_end_package:nn { CJKnumb }
5784 { \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。

```

5785 \bool_if:NT \g__xeCJK_config_bool
5786 {
5787     \ExplSyntaxOff
5788     \file_input:n { \g__xeCJK_config_name_tl .cfg }
5789     \ExplSyntaxOn
5790 }
5791 </package>

```

5.19 xeCJKfntef

```

5792 < *fntef >

5793 \PassOptionsToPackage { normalem } { ulem }
5794 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { ulem } }
5795 \ProcessOptions \scan_stop:

5796 \RequirePackage { xeCJK }
5797 \RequirePackage { ulem }

5798 \addto@hook \UL@hook { \xeCJK_hook_for_ulem: }

\xeCJK_hook_for_ulem: 5799 \cs_new_protected:Npn \xeCJK_hook_for_ulem:
5800 {
5801     \xeCJK_ulem_detect_node:
5802     \l__xeCJK_ulem_text_format_tl
5803     \bool_if:NF \l__xeCJK_ulem_hook_used_bool
5804     {
5805         \bool_set_true:N \l__xeCJK_ulem_hook_used_bool
5806         \__xeCJK_ulem_hook:
5807     }
5808     \xeCJK_ulem_begin_node:
5809 }
5810 \cs_new_protected:Npn \__xeCJK_ulem_hook:
5811 {
5812     \__xeCJK_ulem_initial:
5813     \bool_if:NT \l__xeCJK_ulem_subtract_bool
5814     {
5815         \xeCJK_swap_cs:NN \UL@leaders \xeCJK_ulem_leaders:
5816         \cs_set_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:
5817         \cs_set_eq:NN \xeCJK_ulem_right_skip: \__xeCJK_ulem_right_skip:
5818     }
5819     \bool_if:NT \l__xeCJK_ulem_hidden_bool
5820     { \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_hidden_box: }
5821     \bool_if:NTF \l__xeCJK_ulem_skip_bool
5822     {
5823         \cs_set_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
5824         \cs_set_eq:NN \__xeCJK_ulem_hskip_aux:n \xeCJK_ulem_hskip:n
5825     }

```

```

5826     {
5827         \xeCJK_swap_cs:NN \__xeCJK_punct_hskip:n \__xeCJK_ulem_punct_hskip:n
5828         \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_begin:
5829         \xeCJK_cs_clear:N \__xeCJK_ulem_skip_punct_end:
5830     }
5831     \xeCJK_glue_to_skip:nN
5832     {
5833         \cs_set_eq:NN \ \tex_space:D
5834         \cs_set_eq:NN \penalty \tex_penalty:D
5835         \cs_set_eq:NN \hskip \skip_horizontal:N
5836         \CJKglue
5837     } \l__xeCJK_ccglue_skip
5838     \xeCJK_glue_to_skip:nN
5839     {
5840         \cs_set_eq:NN \ \tex_space:D
5841         \cs_set_eq:NN \penalty \tex_penalty:D
5842         \cs_set_eq:NN \hskip \skip_horizontal:N
5843         \CJKecglue
5844     } \l__xeCJK_ecglue_skip
5845     \xeCJK_glue_to_skip:nN { \xeCJK_space_glue: } \l__xeCJK_space_skip
5846     \cs_set_protected:Npn \CJKglue
5847     { \__xeCJK_ulem_glue:n \l__xeCJK_ccglue_skip }
5848     \cs_set_protected:Npn \CJKecglue
5849     { \__xeCJK_ulem_glue:n \l__xeCJK_ecglue_skip }
5850     \cs_set_protected:Npn \xeCJK_space_glue:
5851     { \__xeCJK_ulem_glue:n \l__xeCJK_space_skip }
5852     \cs_set_eq:NN \xeCJK_punct_node:N \use_none:n
5853     \cs_set_eq:NN \xeCJK_if_last_punct:TF \use_ii:nn
5854     \keys_set:nn { xeCJK / options }
5855     { CheckFullRight = false , xCJKecglue = false }
5856 }
5857 \skip_new:N \l__xeCJK_space_skip
5858 \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

```

5859 \cs_new_protected:Npn \xeCJK_ulem_word:nw #1 ~
5860 {
5861     \__xeCJK_ulem_start:w #1 ~
5862     \exp_after:wN \if_meaning:w \exp_after:wN \UL@end #1
5863     \exp_after:wN \__xeCJK_ulem_end:
5864     \else:
5865         \exp_after:wN \__xeCJK_ulem_loop:nw
5866     \fi:
5867 }
5868 \cs_new_protected:Npn \__xeCJK_ulem_end:
5869 {
5870     \c_group_end_token
5871     \c_group_end_token
5872     \tex_unskip:D \tex_unskip:D \tex_unskip:D
5873     \xeCJK_ulem_right_skip:
5874     \xeCJK_ulem_group_end:
5875     \xeCJK_ulem_right_node:
5876     \tl_gset_eq:NN \g__xeCJK_last_node_tl \g__xeCJK_ulem_saved_last_node_tl
5877     \int_set:Nn \tex_spacefactor:D { \UL@spfactor }
5878 }
5879 \cs_new_protected:Npn \__xeCJK_ulem_loop:nw
5880 {
5881     \reverse_if:N \if_mode_math:
5882     \reverse_if:N \if_dim:w \tex_lastskip:D = \c_zero_dim
5883     \skip_gset_eq:NN \UL@skip \tex_lastskip:D
5884     \tex_unskip:D
5885     \UL@stop \UL@leaders
5886     \fi:
5887     \fi:
5888     \xeCJK_ulem_word:nw \prg_do_nothing:
5889 }
5890 \cs_new_protected:Npn \__xeCJK_ulem_start:w

```

```

5891 { \exp_after:wN \UL@start }
5892 \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:

```

5893 \cs_new_protected:Npn \xeCJK_ulem_left:
5894 {
5895   \xeCJK_ulem_left_node:
5896   \xeCJK_make_group_tag:
5897 }
5898 \cs_new_eq:NN \xeCJK_ulem_left_node: \prg_do_nothing:
5899 \cs_new_protected:Npn \xeCJK_ulem_detect_node:
5900 {
5901   \scan_stop:
5902   \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
5903   {
5904     \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
5905     \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
5906     \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
5907   }
5908   {
5909     \dim_set_eq:NN \l__xeCJK_tmp_dim \tex_lastkern:D
5910     \tex_unkern:D
5911     \dim_compare:nNnTF \tex_lastkern:D = { - \l__xeCJK_tmp_dim }
5912     {
5913       \tex_unkern:D
5914       \cs_set_protected:Npe \xeCJK_ulem_left_node:
5915       {
5916         \tex_kern:D - \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
5917         \tex_kern:D \dim_use:N \l__xeCJK_tmp_dim \exp_stop_f:
5918       }
5919       \cs_set_protected:Npn \xeCJK_ulem_begin_node:
5920       { { \xeCJK_make_node:n { ulem-begin } } }
5921       \cs_set_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
5922     }
5923     {
5924       \tex_kern:D \l__xeCJK_tmp_dim
5925       \xeCJK_cs_clear:N \xeCJK_ulem_left_node:
5926       \xeCJK_cs_clear:N \xeCJK_ulem_begin_node:
5927       \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
5928     }
5929   }
5930 }
5931 \xeCJK_declare_node:n { ulem-begin }
5932 \cs_new_eq:NN \xeCJK_ulem_begin_node: \prg_do_nothing:

```

__xeCJK_ulem_hskip_first:n 如果第一次调用的 \CJKglue 或 \CJKecglue 由下划线中的第一个文字和之前的内容产生, 就
 \xeCJK_ulem_hskip:n 不用画下划线。

```

5933 \cs_new_protected:Npn \__xeCJK_ulem_hskip_first:n #1
5934 {
5935   \xeCJK_if_last_node:nTF { ulem-begin }
5936   {
5937     \xeCJK_remove_node:
5938     \skip_horizontal:n {#1}
5939   }
5940   { \xeCJK_ulem_hskip:n {#1} }
5941   \cs_set_eq:NN \__xeCJK_ulem_hskip:n \xeCJK_ulem_hskip:n
5942 }
5943 \cs_new_eq:NN \__xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_first:n
5944 \cs_new_protected:Npn \xeCJK_ulem_hskip:n #1
5945 { { \skip_set:Nn \UL@skip {#1} \UL@leaders } }

```

\xeCJK_ulem_right: 在下划线最后的位置保存 node。
 \xeCJK_ulem_right_node:

```

5946 \cs_new_protected:Npn \xeCJK_ulem_right:
5947 {
5948   \scan_stop:

```

```

5949 \tl_gset_eq:NN \g__xeCJK_ulem_saved_last_node_tl \g__xeCJK_last_node_tl
5950 \dim_compare:nNnTF \tex_lastkern:D = \c_zero_dim
5951 { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
5952 {
5953   \dim_compare:nNnTF \tex_lastkern:D = { 3sp }
5954   { \xeCJK_cs_gclear:N \xeCJK_ulem_right_node: }
5955   {
5956     \exp_args:NNo \tex_unkern:D
5957     \__xeCJK_ulem_right_aux:n { \dim_use:N \tex_lastkern:D }
5958   }
5959 }
5960 }
5961 \tl_new:N \g__xeCJK_ulem_saved_last_node_tl
5962 \cs_new_protected:Npn \__xeCJK_ulem_right_aux:n #1
5963 {
5964   \dim_compare:nNnTF \tex_lastkern:D = { - #1 }
5965   {
5966     \tex_unkern:D
5967     \cs_gset_protected:Npn \xeCJK_ulem_right_node:
5968     {
5969       \tex_kern:D - #1 \exp_stop_f:
5970       \tex_kern:D #1 \exp_stop_f:
5971     }
5972     \tl_gset:Ne \UL@spfactor { \int_use:N \tex_spacefactor:D }
5973   }
5974   {
5975     \tex_kern:D #1 \exp_stop_f:
5976     \xeCJK_cs_gclear:N \xeCJK_ulem_right_node:
5977   }
5978 }
5979 \cs_new_eq:NN \xeCJK_ulem_right_node: \prg_do_nothing:

```

\xeCJK_ulem_var_leaders: 第一次画下划线时, 不需要向左平移 \UL@pixel, 让左侧有间距。

```

5980 \cs_new_protected:Npn \xeCJK_ulem_leaders:
5981 { \__xeCJK_ulem_var_leaders: }
5982 \cs_new_protected:Npn \xeCJK_ulem_var_leaders:
5983 {
5984   \scan_stop:
5985   \skip_if_eq:nnF { \UL@skip } { \c_zero_skip }
5986   {
5987     \UL@leadtype \skip_horizontal:n { \UL@skip + \UL@pixel }
5988     \skip_horizontal:n { - \UL@pixel }
5989     \cs_gset_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_leaders:
5990   }
5991 }
5992 \cs_new_eq:NN \__xeCJK_ulem_var_leaders: \xeCJK_ulem_var_leaders:

```

\xeCJK_ulem_right_skip: 在下划线完全画好之后, 我们检测最后的情况。用 \unskip 去掉最后一个下划线, 再重新画一个减少 \UL@pixel 的。

```

5993 \cs_new_eq:NN \xeCJK_ulem_right_skip: \prg_do_nothing:
5994 \cs_new_protected:Npn \__xeCJK_ulem_right_skip:
5995 {
5996   \int_case:nn { \tex_lastnodetype:D }
5997   {
5998     { \c__xeCJK_hlist_node } { \__xeCJK_ulem_right_skip_hbox: }
5999     { \c__xeCJK_glue_node } { \__xeCJK_ulem_right_skip_glue: }
6000     { \c__xeCJK_penalty_node } { \__xeCJK_ulem_right_skip_penalty: }
6001   }
6002 }
6003 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_hbox:
6004 {
6005   \box_set_to_last:N \l__xeCJK_tmp_box
6006   \__xeCJK_if_last_kern:TF
6007   { \__xeCJK_ulem_right_skip_kern: }
6008   { \__xeCJK_ulem_right_skip_glue: }
6009   \box_use_drop:N \l__xeCJK_tmp_box

```

```

6010 }
6011 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_kern:
6012 {
6013   \dim_set:Nn \l__xeCJK_tmp_dim { - \box_wd:N \l__xeCJK_tmp_box }
6014   \dim_compare:nNnT \tex_lastkern:D = \l__xeCJK_tmp_dim
6015   {
6016     \tex_unkern:D
6017     \__xeCJK_ulem_right_skip_glue:
6018     \tex_kern:D \l__xeCJK_tmp_dim
6019   }
6020 }
6021 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_glue:
6022 {
6023   \skip_if_eq:nnT { \tex_lastskip:D } { - \UL@pixel }
6024   {
6025     \tex_unskip:D
6026     \skip_set:Nn \l__xeCJK_tmp_skip { \tex_lastskip:D - \UL@pixel }
6027     \tex_unskip:D
6028     \UL@leadtype \skip_horizontal:N \l__xeCJK_tmp_skip
6029   }
6030 }
6031 \cs_new_protected:Npn \__xeCJK_ulem_right_skip_penalty:
6032 {
6033   \int_set_eq:NN \l__xeCJK_tmp_int \tex_lastpenalty:D
6034   \tex_unpenalty:D
6035   \__xeCJK_if_last_hlist:T
6036   { \__xeCJK_ulem_right_skip_hbox: }
6037   \tex_penalty:D \l__xeCJK_tmp_int
6038 }

```

__xeCJK_ulem_hidden_box: 只画线, 不输出盒子。

```

6039 \cs_new_protected:Npn \__xeCJK_ulem_hidden_box:
6040 {
6041   \tl_if_empty:NF \UL@start
6042   {
6043     \box_set_ht:Nn \l__xeCJK_hidden_box { \box_ht:N \UL@box }
6044     \box_set_dp:Nn \l__xeCJK_hidden_box { \box_dp:N \UL@box }
6045     \box_use:N \l__xeCJK_hidden_box
6046     \xeCJK_no_break:
6047     \xeCJK_ulem_hskip:n { \box_wd:N \UL@box }
6048     \box_use:N \l__xeCJK_hidden_box
6049   }
6050 }
6051 \box_new:N \l__xeCJK_hidden_box
6052 \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:

```

6053 \cs_new_protected:Npn \__xeCJK_ulem_skip_punct_begin:
6054 {
6055   \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_skip_putbox:
6056   \cs_set_eq:NN \xeCJK_ulem_hskip:n \skip_horizontal:n
6057 }
6058 \cs_new_protected:Npn \__xeCJK_ulem_skip_punct_end:
6059 {
6060   \cs_set_eq:NN \UL@putbox \__xeCJK_ulem_putbox:
6061   \cs_set_eq:NN \xeCJK_ulem_hskip:n \__xeCJK_ulem_hskip_aux:n
6062 }
6063 \cs_new_eq:NN \__xeCJK_ulem_putbox: \UL@putbox
6064 \cs_new_protected:Npn \__xeCJK_ulem_skip_putbox:
6065 {
6066   \tl_if_empty:NF \UL@start
6067   { \box_use_drop:N \UL@box }
6068 }

```

`\_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。

```

6069 \cs_new_protected:Npn \_xeCJK_ulem_initial:
6070 {
6071   \_xeCJK_ulem_swap_cs:NN
6072   \xeCJK_FullLeft_and_Default: \_xeCJK_ulem_FullLeft_and_Default:
6073   \xeCJK_FullLeft_and_CJK: \_xeCJK_ulem_FullLeft_and_CJK:
6074   \xeCJK_FullLeft_and_Boundary: \_xeCJK_ulem_FullLeft_and_Boundary:
6075   \xeCJK_FullRight_and_Default: \_xeCJK_ulem_FullRight_and_Default:
6076   \xeCJK_FullRight_and_CJK: \_xeCJK_ulem_FullRight_and_CJK:
6077   \xeCJK_FullRight_and_CJStarter: \_xeCJK_ulem_FullRight_and_CJStarter:
6078   \xeCJK_FullRight_and_Boundary: \_xeCJK_ulem_FullRight_and_Boundary:
6079   \xeCJK_CJK_and_CJK:N \_xeCJK_ulem_CJK_and_CJK:N
6080   \xeCJK_CJK_and_Boundary:w \_xeCJK_ulem_CJK_and_Boundary:w
6081   \xeCJK@fix@penalty \_xeCJK_ulem_fix_penalty:
6082   \_xeCJK_punct_breakable_kern:n \_xeCJK_ulem_punct_breakable_kern:n
6083   \_xeCJK_Default_and_FullLeft_glue:N \_xeCJK_ulem_Default_and_FullLeft_glue:N
6084   \_xeCJK_Default_and_FullRight_glue:N \_xeCJK_ulem_Default_and_FullRight_glue:N
6085   \_xeCJK_CJK_and_FullLeft_glue:N \_xeCJK_ulem_CJK_and_FullLeft_glue:N
6086   \_xeCJK_CJK_and_FullRight_glue:N \_xeCJK_ulem_CJK_and_FullRight_glue:N
6087   \_xeCJK_Boundary_and_FullLeft_glue:N \_xeCJK_ulem_Boundary_and_FullLeft_glue:N
6088   \q_recursion_tail \q_nil \q_recursion_stop
6089   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
6090   {
6091     \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
6092     {
6093       \str_if_eq:nnTF {##1} {####1}
6094       {
6095         \xeCJK_inter_class_toks:nnn { CJK } { CJK/##1 }
6096         { \_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
6097         \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/##1 }
6098         { \_xeCJK_ulem_between_CJK_blocks:nnN { CJK } {##1} }
6099       }
6100       {
6101         \xeCJK_inter_class_toks:nnn { CJK/##1 } { CJK/####1 }
6102         { \_xeCJK_ulem_between_CJK_blocks:nnN {##1} {####1} }
6103       }
6104     }
6105   }
6106 }
6107 \cs_new_protected:Npn \_xeCJK_ulem_swap_cs:NN #1#2
6108 {
6109   \quark_if_recursion_tail_stop:N #1
6110   \xeCJK_swap_cs:NN #1#2
6111   \_xeCJK_ulem_swap_cs:NN
6112 }

```

`\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。

```

6113 \cs_new:Npn \xeCJK_if_ulem_patch:TF
6114 {
6115   \if_meaning:w \_ \LA@space
6116   \exp_after:wN \use_ii:nn
6117   \else:
6118     \exp_after:wN \use_i:nn
6119   \fi:
6120 }

```

```

\_xeCJK_ulem_CJK_and_Boundary:w 6121 \cs_new_protected:Npn \_xeCJK_ulem_CJK_and_Boundary:w
6122 {
6123   \xeCJK_if_ulem_patch:TF
6124   {
6125     \xeCJK_peek_catcode_ignore_spaces:NTF \c_math_toggle_token
6126     { \xeCJK_class_group_end: \CJKeclue }

```

```

6127         {
6128             \bool_if:NTF \l__xeCJK_peek_ignore_spaces_bool
6129             { \__xeCJK_ulem_peek_math:w }
6130             { \__xeCJK_ulem_group_end:n { CJK } }
6131         }
6132     }
6133     { \__xeCJK_ulem_CJK_and_Boundary:w }
6134 }
6135 \cs_new_protected:Npn \__xeCJK_ulem_group_end:n #1
6136 {
6137     \xeCJK_class_group_end: \UL@stop
6138     \UL@start { \xeCJK_make_node:n {#1} }
6139     \xeCJK_make_group_tag:
6140     \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
6141 }

```

__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。

```

6142 \cs_new_protected:Npn \__xeCJK_ulem_peek_math:w
6143 {
6144     \cs_set_eq:NN \__xeCJK_ulem_start:w \__xeCJK_ulem_exp_stop:w
6145     \exp_after:wN \peek_after:Nw
6146     \exp_after:wN \__xeCJK_ulem_peek_math_branches:w
6147     \exp:w \exp_end_continue_f:w
6148 }
6149 \cs_new_protected:Npn \__xeCJK_ulem_peek_math_branches:w
6150 {
6151     \token_if_math_toggle:NTF \l_peek_token
6152     { \xeCJK_class_group_end: \CJKe glue }
6153     { \__xeCJK_ulem_group_end:n { CJK-space } }
6154 }
6155 \cs_new_protected:Npn \__xeCJK_ulem_exp_stop:w
6156 {
6157     \cs_if_eq:NNTF \UL@start \@empty
6158     { \exp_after:wN \exp_stop_f: }
6159     { \exp_after:wN \UL@start }
6160 }

```

__xeCJK_ulem_fix_penalty: 6161 \cs_new_protected:Npn __xeCJK_ulem_fix_penalty:

```

6162 {
6163     \xeCJK_if_ulem_patch:TF
6164     { \fix@penalty }
6165     { \__xeCJK_ulem_fix_penalty: }
6166 }

```

__xeCJK_ulem_save_font_state: 在 \xeCJK_class_group_end: 前保存字体状态, 在 __xeCJK_ulem_class_group_begin: 后恢复, 使 \bfseries、\sffamily 等字体声明能跨越下划线内的 CJK 分组边界。

```

\__xeCJK_ulem_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
6167 \tl_new:N \g__xeCJK_ulem_saved_series_tl
6168 \tl_new:N \g__xeCJK_ulem_saved_shape_tl
6169 \tl_new:N \g__xeCJK_ulem_saved_CJK_family_tl
6170 \tl_new:N \g__xeCJK_ulem_saved_CJK_fam_tl
6171 \cs_new_protected:Npn \__xeCJK_ulem_save_font_state:
6172 {
6173     \tl_gset:Ne \g__xeCJK_ulem_saved_series_tl { \f@series }
6174     \tl_gset:Ne \g__xeCJK_ulem_saved_shape_tl { \f@shape }
6175     \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_family_tl \l_xeCJK_family_tl
6176     \tl_gset_eq:NN \g__xeCJK_ulem_saved_CJK_fam_tl \CJK@family
6177 }
6178 \cs_new_protected:Npn \__xeCJK_ulem_restore_font_state:
6179 {
6180     \tl_set:Ne \f@series { \g__xeCJK_ulem_saved_series_tl }
6181     \tl_set:Ne \f@shape { \g__xeCJK_ulem_saved_shape_tl }
6182     \tl_set_eq:NN \l_xeCJK_family_tl \g__xeCJK_ulem_saved_CJK_family_tl
6183     \tl_set_eq:NN \CJK@family \g__xeCJK_ulem_saved_CJK_fam_tl
6184 }

```

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

```

\__xeCJK_ulem_CJK_and_CJK:N 6185 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_CJK:N
6186 {
6187   \xeCJK_if_ulem_patch:TF
6188   {
6189     \__xeCJK_ulem_save_font_state:
6190     \xeCJK_class_group_end:
6191     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
6192     \__xeCJK_ulem_class_group_begin:
6193     \__xeCJK_ulem_restore_font_state:
6194     \xeCJK_select_font:
6195     \xeCJK_fallback_symbol:NN
6196     \CJKsymbol
6197   }
6198   { \__xeCJK_ulem_CJK_and_CJK:N }
6199 }

\__xeCJK_ulem_class_group_begin: 6200 \cs_new_protected:Npn \__xeCJK_ulem_class_group_begin:
6201 {
6202   \xeCJK_class_group_begin:
6203   \xeCJK_clear_Boundary_and_CJK_toks:
6204 }

\__xeCJK_ulem_between_CJK_blocks:nnN 6205 \cs_new_protected:Npn \__xeCJK_ulem_between_CJK_blocks:nnN #1#2
6206 {
6207   \xeCJK_if_ulem_patch:TF
6208   {
6209     \__xeCJK_ulem_save_font_state:
6210     \xeCJK_class_group_end:
6211     \UL@stop \__xeCJK_ulem_ccglue: \UL@start
6212     \xeCJK_class_group_begin:
6213     \xeCJK_clear_Boundary_and_CJK_toks:
6214     \__xeCJK_ulem_restore_font_state:
6215     \__xeCJK_switch_font:nn {#1} {#2}
6216     \xeCJK_fallback_symbol:NN
6217     \CJKsymbol
6218   }
6219   {
6220     \skip_horizontal:N \l__xeCJK_ccglue_skip
6221     \__xeCJK_switch_font:nn {#1} {#2}
6222     \xeCJK_fallback_symbol:NN
6223     \CJKsymbol
6224   }
6225 }

\__xeCJK_ulem_Default_and_FullLeft_glue:N 6226 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullLeft_glue:N #1
6227 {
6228   \xeCJK_if_ulem_patch:TF
6229   {
6230     \UL@stop
6231     \__xeCJK_ulem_skip_punct_begin:
6232     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
6233     \UL@start
6234   }
6235   { \__xeCJK_ulem_Default_and_FullLeft_glue:N #1 }
6236 }

\__xeCJK_ulem_Boundary_and_FullLeft_glue:N 6237 \cs_new_protected:Npn \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1
6238 {
6239   \xeCJK_if_ulem_patch:TF
6240   {
6241     \UL@stop
6242     \__xeCJK_ulem_skip_punct_begin:
6243     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
6244     \UL@start
6245   }
6246   { \__xeCJK_ulem_Boundary_and_FullLeft_glue:N #1 }
6247 }

```

```

\__xeCJK_ulem_CJK_and_FullLeft_glue:N 6248 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1
6249 {
6250   \xeCJK_if_ulem_patch:TF
6251   {
6252     \xeCJK_class_group_end:
6253     \UL@stop
6254     \__xeCJK_ulem_skip_punct_begin:
6255     \__xeCJK_ulem_punct_ccglue:
6256     \__xeCJK_punct_glue:NN \c__xeCJK_left_tl #1
6257     \UL@start
6258     \__xeCJK_ulem_class_group_begin:
6259     \xeCJK_select_punct_font:
6260   }
6261   { \__xeCJK_ulem_CJK_and_FullLeft_glue:N #1 }
6262 }

\__xeCJK_ulem_Default_and_FullRight_glue:N 6263 \cs_new_protected:Npn \__xeCJK_ulem_Default_and_FullRight_glue:N #1
6264 {
6265   \xeCJK_if_ulem_patch:TF
6266   {
6267     \UL@stop
6268     \__xeCJK_ulem_skip_punct_begin:
6269     \__xeCJK_punct_if_long:NTF #1
6270     { \xeCJK_allow_break: }
6271     { \xeCJK_no_break: }
6272     \__xeCJK_punct_if_middle:NT #1
6273     {
6274       \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
6275       \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
6276     }
6277     \UL@start
6278   }
6279   { \__xeCJK_ulem_Default_and_FullRight_glue:N #1 }
6280 }

\__xeCJK_ulem_CJK_and_FullRight_glue:N 6281 \cs_new_protected:Npn \__xeCJK_ulem_CJK_and_FullRight_glue:N #1
6282 {
6283   \xeCJK_if_ulem_patch:TF
6284   {
6285     \xeCJK_class_group_end:
6286     \UL@stop
6287     \__xeCJK_ulem_skip_punct_begin:
6288     \__xeCJK_punct_if_long:NTF #1
6289     { \xeCJK_allow_break: }
6290     { \xeCJK_no_break: }
6291     \__xeCJK_punct_if_middle:NT #1
6292     {
6293       \__xeCJK_ulem_punct_ccglue:
6294       \__xeCJK_punct_glue:NN \c__xeCJK_right_tl #1
6295       \__xeCJK_punct_rule:NN \c__xeCJK_left_tl #1
6296     }
6297     \UL@start
6298     \__xeCJK_ulem_class_group_begin:
6299     \xeCJK_select_punct_font:
6300   }
6301   { \__xeCJK_ulem_CJK_and_FullRight_glue:N #1 }
6302 }

\__xeCJK_ulem_FullLeft_and_Default: 6303 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_Default:
6304 {
6305   \xeCJK_if_ulem_patch:TF
6306   {
6307     \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
6308     {
6309       \xeCJK_get_punct_bounds:No \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
6310       \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6311       \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
6312       \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl

```

```

6313     }
6314     { \xeCJK_class_group_end: \UL@stop }
6315     \__xeCJK_ulem_skip_punct_end:
6316     \xeCJK_no_break:
6317     \UL@start
6318   }
6319   { \__xeCJK_ulem_FullLeft_and_Default: }
6320 }

\_xeCJK_ulem_FullLeft_and_Boundary: 6321 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_Boundary:
6322 {
6323   \xeCJK_if_ulem_patch:TF
6324   {
6325     \__xeCJK_punct_if_middle:NTF \g__xeCJK_last_punct_tl
6326     {
6327       \xeCJK_get_punct_bounds:No \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
6328       \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6329       \xeCJK_class_group_end: \UL@stop \xeCJK_no_break:
6330       \__xeCJK_punct_glue:NN \c__xeCJK_left_tl \g__xeCJK_last_punct_tl
6331     }
6332     { \xeCJK_class_group_end: \UL@stop }
6333     \__xeCJK_ulem_skip_punct_end:
6334     \xeCJK_no_break:
6335     \UL@start
6336     \tex_ignorespaces:D
6337   }
6338   { \__xeCJK_ulem_FullLeft_and_Boundary: }
6339 }

\_xeCJK_ulem_FullLeft_and_CJK: 6340 \cs_new_protected:Npn \__xeCJK_ulem_FullLeft_and_CJK:
6341 {
6342   \xeCJK_if_ulem_patch:TF
6343   {
6344     \xeCJK_FullLeft_and_Default:
6345     \__xeCJK_ulem_class_group_begin:
6346     \xeCJK_select_font:
6347   }
6348   { \__xeCJK_ulem_FullLeft_and_CJK: }
6349 }

\_xeCJK_ulem_FullRight_and_Default: 6350 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Default:
6351 {
6352   \xeCJK_if_ulem_patch:TF
6353   {
6354     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6355     \xeCJK_class_group_end:
6356     \UL@stop
6357     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6358     \__xeCJK_ulem_skip_punct_end:
6359     \UL@start
6360   }
6361   { \__xeCJK_ulem_FullRight_and_Default: }
6362 }

\_xeCJK_ulem_FullRight_and_Boundary: 6363 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_Boundary:
6364 {
6365   \xeCJK_if_ulem_patch:TF
6366   {
6367     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6368     \xeCJK_class_group_end:
6369     \UL@stop
6370     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6371     \__xeCJK_ulem_skip_punct_end:
6372     \UL@start
6373     \tex_ignorespaces:D
6374   }
6375   { \__xeCJK_ulem_FullRight_and_Boundary: }
6376 }

```

```

\__xeCJK_ulem_FullRight_and_CJK: 6377 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJK:
6378 {
6379   \xeCJK_if_ulem_patch:TF
6380   {
6381     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6382     \xeCJK_class_group_end:
6383     \UL@stop
6384     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6385     \__xeCJK_ulem_punct_ccglue:
6386     \__xeCJK_ulem_skip_punct_end:
6387     \UL@start
6388     \__xeCJK_ulem_class_group_begin:
6389     \xeCJK_select_font:
6390   }
6391   { \__xeCJK_ulem_FullRight_and_CJK: }
6392 }

```

__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。

```

6393 \cs_new_protected:Npn \__xeCJK_ulem_FullRight_and_CJStarter:
6394 {
6395   \xeCJK_if_ulem_patch:TF
6396   {
6397     \__xeCJK_punct_rule:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6398     \xeCJK_class_group_end:
6399     \UL@stop
6400     \xeCJK_no_break:
6401     \__xeCJK_punct_glue:NN \c__xeCJK_right_tl \g__xeCJK_last_punct_tl
6402     \__xeCJK_ulem_punct_ccglue:
6403     \__xeCJK_ulem_skip_punct_end:
6404     \UL@start
6405     \__xeCJK_ulem_class_group_begin:
6406     \xeCJK_select_font:
6407   }
6408   { \__xeCJK_ulem_FullRight_and_CJStarter: }
6409 }

```

```

\__xeCJK_ulem_punct_hskip:n 6410 \cs_new_protected:Npn \__xeCJK_ulem_punct_hskip:n
6411 {
6412   \xeCJK_if_ulem_patch:TF
6413   { \xeCJK_ulem_hskip:n }
6414   { \__xeCJK_ulem_punct_hskip:n }
6415 }

```

```

\__xeCJK_ulem_punct_breakable_kern:n 6416 \cs_new_protected:Npn \__xeCJK_ulem_punct_breakable_kern:n #1
6417 {
6418   \xeCJK_if_ulem_patch:TF
6419   {
6420     \xeCJK_class_group_end:
6421     \UL@stop \xeCJK_ulem_hskip:n {#1} \UL@start
6422     \__xeCJK_ulem_class_group_begin:
6423     \xeCJK_select_punct_font:
6424   }
6425   { \__xeCJK_ulem_punct_breakable_kern:n {#1} }
6426 }

```

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

```

\__xeCJK_ulem_ccglue: 6427 \cs_new_protected:Npn \__xeCJK_ulem_glue:n #1
\__xeCJK_ulem_punct_ccglue: 6428 {
6429   \xeCJK_if_ulem_patch:TF
6430   {
6431     \tl_if_empty:NTF \l__xeCJK_group_tag_tl
6432     { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }
6433     {
6434       \str_if_eq:eeTF { \l__xeCJK_group_tag_tl } { \c__xeCJK_group_tag_tl }
6435       { \UL@stop \__xeCJK_ulem_hskip:n {#1} \UL@start }

```

```

6436             { \skip_horizontal:n {#1} }
6437         }
6438     }
6439     { \skip_horizontal:n {#1} }
6440 }
6441 \cs_new_protected:Npn \xeCJK_make_group_tag:
6442 { { \tl_set:Nx \l__xeCJK_group_tag_tl { \c__xeCJK_group_tag_tl } }
6443 \tl_new:N \l__xeCJK_group_tag_tl
6444 \tl_const:Nn \c__xeCJK_group_tag_tl
6445 {
6446     T \int_use:N \tex_currentgrouptype:D
6447     L \int_use:N \tex_currentgrouplevel:D
6448 }
6449 \cs_new_protected:Npn \__xeCJK_ulem_ccglue:
6450 { { \skip_set_eq:NN \UL@skip \l__xeCJK_ccglue_skip \UL@leaders } }
6451 \cs_new_protected:Npn \__xeCJK_ulem_punct_ccglue:
6452 { \__xeCJK_punct_hskip:n { \l__xeCJK_ccglue_skip } }

\xeCJK_ulem_group_begin: 6453 \cs_new_protected:Npn \xeCJK_ulem_group_begin:
\xeCJK_ulem_group_end:   6454 {
\xeCJK_ulem_on:n         6455     \mode_leave_vertical:
6456     \c_group_begin_token
6457 }
6458 \cs_new_protected:Npn \xeCJK_ulem_group_end:
6459 { \c_group_end_token }
6460 \cs_new_protected:Npn \xeCJK_ulem_on:n
6461 { \ULon }
6462 \cs_new_eq:NN \__xeCJK_ulem_on:n \UL@on
6463 \cs_set_protected:Npn \UL@on #1
6464 { \__xeCJK_ulem_on:n { \xeCJK_ulem_left: #1 \xeCJK_ulem_right: } }

```

\xeCJKfntefon 扩展 \ULon 的参数。

```

6465 \NewDocumentCommand \xeCJKfntefon { s t- s o }
6466 {
6467     \mode_leave_vertical:
6468     \xeCJK_ulem_boot:NNNn #1#2#3 {#4}
6469     \xeCJK_ulem_on:n
6470 }

```

\CJKKunderline

```

6471 \NewDocumentCommand \CJKKunderline { s t- s o }
6472 {
6473     \xeCJK_ulem_group_begin:
6474     \xeCJK_fntef_boot:nnNNNn { underline } { uline } #1#2#3 {#4}
6475     \cs_set_eq:NN \ULleaders \tex_cleaders:D
6476     \xeCJK_fntef_initial:nnn
6477     { \l__xeCJK_uline_depth_tl }
6478     { \l__xeCJK_uline_sep_tl }
6479     {
6480         \l__xeCJK_uline_format_tl
6481         \tex_vrule:D
6482         height \dim_eval:n { \l__xeCJK_uline_thickness_tl }
6483         depth \c_zero_dim
6484         width .2em
6485     }
6486     \xeCJK_ulem_on:n
6487 }
6488 \NewDocumentCommand \varCJKKunderline { }
6489 { \CJKKunderline - }

```

\CJKKunderwave

```

6490 \NewDocumentCommand \CJKKunderwave { s t- s o }
6491 {
6492     \xeCJK_ulem_group_begin:
6493     \xeCJK_fntef_boot:nnNNNn { underwave } { uwave } #1#2#3 {#4}
6494     \cs_set_eq:NN \ULleaders \tex_xleaders:D
6495     \xeCJK_fntef_initial:nnn
6496     { \l__xeCJK_uwave_depth_tl }
6497     { \l__xeCJK_uwave_sep_tl }

```

```

6498     { \l__xeCJK_uwave_format_tl \l__xeCJK_uwave_symbol_tl }
6499     \xeCJK_ulem_on:n
6500 }

\CJKunderdblline 6501 \NewDocumentCommand \CJKunderdblline { s t- s o }
6502 {
6503     \xeCJK_ulem_group_begin:
6504     \xeCJK_fntef_boot:nnNNNn { underdblline } { udbline } #1#2#3 {#4}
6505     \cs_set_eq:NN \ULleaders \tex_cleaders:D
6506     \xeCJK_fntef_initial:nnn
6507     { \l__xeCJK_udbline_depth_tl }
6508     { \l__xeCJK_udbline_sep_tl }
6509     {
6510         \l__xeCJK_udbline_format_tl
6511         \vbox_top:n
6512         {
6513             \tex_hruler:D
6514             height \dim_eval:n { \l__xeCJK_udbline_thickness_tl }
6515             depth \c_zero_dim
6516             width .2em
6517             \tex_kern:D \dim_eval:n { \l__xeCJK_udbline_gap_tl }
6518             \tex_hruler:D
6519             height \dim_eval:n { \l__xeCJK_udbline_thickness_tl }
6520             depth \c_zero_dim
6521             width .2em
6522         }
6523     }
6524     \xeCJK_ulem_on:n
6525 }

\CJKsout 6526 \NewDocumentCommand \CJKsout { s t- s o }
6527 {
6528     \xeCJK_ulem_group_begin:
6529     \xeCJK_fntef_boot:nnNNNn { sout } { sout } #1#2#3 {#4}
6530     \cs_set_eq:NN \ULleaders \tex_cleaders:D
6531     \xeCJK_fntef_initial:nn
6532     {
6533         \l__xeCJK_sout_format_tl
6534         \tex_vruler:D
6535         height \dim_eval:n { \l__xeCJK_sout_thickness_tl }
6536         depth \c_zero_dim
6537         width .2em
6538     }
6539     {
6540         \box_move_up:nn
6541         { \l__xeCJK_sout_height_tl - \box_ht:N \l__xeCJK_fntef_box / 2 }
6542         { \box_use:N \l__xeCJK_fntef_box }
6543     }
6544     \xeCJK_ulem_on:n
6545 }

\CJKxout 6546 \NewDocumentCommand \CJKxout { s t- s o }
6547 {
6548     \xeCJK_ulem_group_begin:
6549     \xeCJK_fntef_boot:nnNNNn { xout } { xout } #1#2#3 {#4}
6550     \cs_set_eq:NN \ULleaders \tex_cleaders:D
6551     \xeCJK_fntef_initial:nn
6552     {
6553         \l__xeCJK_xout_format_tl
6554         \tex_kern:D -.1 em $/$
6555         \tex_kern:D -.1 em
6556     }
6557     {
6558         \box_move_up:nn
6559         { \box_dp:N \l__xeCJK_fntef_box / 2 }
6560         { \box_use:N \l__xeCJK_fntef_box }
6561     }
6562     \xeCJK_ulem_on:n

```

```

6563 }
\CJKunderanyline 6564 \NewDocumentCommand \CJKunderanyline { s t- s o m m }
6565 {
6566   \xeCJK_ulem_group_begin:
6567     \xeCJK_ulem_boot:NNNn #1#2#3 {#4}
6568     \cs_set_eq:NN \ULleaders \tex_cleaders:D
6569     \xeCJK_fntef_initial:nn
6570       {#6}
6571       {
6572         \box_move_down:nn
6573         {#5}
6574         { \box_use:N \l__xeCJK_fntef_box }
6575       }
6576     \tl_if_empty:NF \l__xeCJK_ulem_boxdepth_tl
6577     { \box_set_dp:Nn \ULC@box { \l__xeCJK_ulem_boxdepth_tl } }
6578     \tl_if_empty:NF \l__xeCJK_ulem_sep_tl
6579     {
6580       \bool_set_true:N \l__xeCJK_fntef_bool
6581       \dim_set:Nn \l__xeCJK_fntef_dim
6582       { \l__xeCJK_ulem_sep_tl + \box_dp:N \ULC@box }
6583     }
6584     \xeCJK_ulem_on:n
6585   }

```

\xeCJK_fntef_boot:nnNNNn 处理参数问题。

```

6586 \cs_new_protected:Npn \xeCJK_fntef_boot:nnNNNn #1#2#3#4#5#6
6587 {
6588   \bool_lazy_or:nnT {#3} {#5}
6589   { \bool_set_false:c { l__xeCJK_#2_skip_bool } }
6590   \bool_if:NT #4
6591   { \bool_set_true:c { l__xeCJK_#2_subtract_bool } }
6592   \tl_if_novalue:nF {#6}
6593   { \keys_set:nn { xeCJK / options / #1 } {#6} }
6594   \bool_set_eq:Nc \l__xeCJK_ulem_skip_bool { l__xeCJK_#2_skip_bool }
6595   \bool_set_eq:Nc \l__xeCJK_ulem_hidden_bool { l__xeCJK_#2_hidden_bool }
6596   \bool_set_eq:Nc \l__xeCJK_ulem_subtract_bool { l__xeCJK_#2_subtract_bool }
6597   \tl_set_eq:Nc \l__xeCJK_ulem_text_format_tl { l__xeCJK_#2_text_format_tl }
6598 }
6599 \cs_new_protected:Npn \xeCJK_ulem_boot:NNNn #1#2#3#4
6600 {
6601   \bool_lazy_or:nnT {#1} {#3}
6602   { \bool_set_false:N \l__xeCJK_ulem_skip_bool }
6603   \bool_if:NT #2
6604   { \bool_set_true:N \l__xeCJK_ulem_subtract_bool }
6605   \tl_if_novalue:nF {#4}
6606   { \keys_set:nn { xeCJK / options / ulem } {#4} }
6607 }

```

\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。

```

6608 \cs_new_protected:Npn \xeCJK_fntef_initial:n
6609 {
6610   \bool_if:NFT \l__xeCJK_nest_bool
6611   { \__xeCJK_warning:n { fntef-nesting } }
6612   {
6613     \bool_set_true:N \l__xeCJK_nest_bool
6614     \__xeCJK_restore_shipout_fntef:
6615   }
6616   \xeCJK_fntef_sbox:n
6617 }
6618 \cs_new_protected:Npn \xeCJK_fntef_initial:nn #1
6619 {
6620   \xeCJK_fntef_initial:n {#1}
6621   \bool_if:NFT \l__xeCJK_fntef_bool
6622   { \dim_zero:N \l__xeCJK_fntef_dim }
6623   \markoverwith

```

```

6624 }
6625 \cs_new_protected:Npn \xeCJK_fntef_initial:nnn #1#2#3
6626 {
6627   \xeCJK_fntef_initial:n {#3}
6628   \bool_if:NF \l__xeCJK_fntef_bool
6629   {
6630     \bool_set_true:N \l__xeCJK_fntef_bool
6631     \dim_set:Nn \l__xeCJK_fntef_dim {#1}
6632   }
6633   \markoverwith
6634   {
6635     \box_move_down:nn
6636     { \l__xeCJK_fntef_dim + \box_ht:N \l__xeCJK_fntef_box }
6637     { \box_use:N \l__xeCJK_fntef_box }
6638   }
6639   \dim_set:Nn \l__xeCJK_fntef_dim { #2 + \box_dp:N \ULC@box }
6640 }
6641 \box_new:N \l__xeCJK_fntef_box
6642 \cs_new_eq:NN \xeCJKfntefbox \l__xeCJK_fntef_box
6643 \bool_new:N \l__xeCJK_nest_bool
6644 \bool_new:N \l__xeCJK_fntef_bool
6645 \__xeCJK_msg_new:nn { fntef-nesting }
6646 { 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。

```

6647 \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。

```

6648 \cs_new_protected:Npn \xeCJK_fntef_sbox:n #1
6649 {
6650   \tl_set_eq:NN \l__xeCJK_tmp_tl \g__xeCJK_last_node_tl
6651   \hbox_set:Nn \l__xeCJK_fntef_box
6652   { \color_ensure_current: #1 }
6653   \tl_gset_eq:NN \g__xeCJK_last_node_tl \l__xeCJK_tmp_tl
6654 }

```

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。

```

6655 \keys_define:nn { xeCJK / options }
6656 {
6657   underdot / symbol      .tl_set:N = \l__xeCJK_udot_symbol_tl ,
6658   underdot / depth      .tl_set:N = \l__xeCJK_udot_depth_tl ,
6659   underdot / sep        .tl_set:N = \l__xeCJK_udot_sep_tl ,
6660   underdot / format     .tl_set:N = \l__xeCJK_udot_format_tl ,
6661   underdot / textformat .tl_set:N = \l__xeCJK_udot_text_format_tl ,
6662   underdot / boxdepth   .tl_set:N = \l__xeCJK_udot_boxdepth_tl ,
6663   symbol / sep          .tl_set:N = \l__xeCJK_symbol_sep_tl ,
6664   symbol / boxdepth     .tl_set:N = \l__xeCJK_symbol_boxdepth_tl ,
6665   symbol / textformat   .tl_set:N = \l__xeCJK_symbol_text_format_tl ,
6666   underline / skip      .bool_set:N = \l__xeCJK_uline_skip_bool ,
6667   underline / hidden    .bool_set:N = \l__xeCJK_uline_hidden_bool ,
6668   underline / subtract  .bool_set:N = \l__xeCJK_uline_subtract_bool ,
6669   underline / thickness .tl_set:N = \l__xeCJK_uline_thickness_tl ,
6670   underline / depth     .tl_set:N = \l__xeCJK_uline_depth_tl ,
6671   underline / sep       .tl_set:N = \l__xeCJK_uline_sep_tl ,
6672   underline / format    .tl_set:N = \l__xeCJK_uline_format_tl ,
6673   underline / textformat .tl_set:N = \l__xeCJK_uline_text_format_tl ,
6674   underdblline / skip   .bool_set:N = \l__xeCJK_udbline_skip_bool ,
6675   underdblline / hidden .bool_set:N = \l__xeCJK_udbline_hidden_bool ,
6676   underdblline / subtract .bool_set:N = \l__xeCJK_udbline_subtract_bool ,
6677   underdblline / thickness .tl_set:N = \l__xeCJK_udbline_thickness_tl ,

```

```

6678   underdblline / depth      .tl_set:N = \l__xeCJK_udbline_depth_tl ,
6679   underdblline / sep        .tl_set:N = \l__xeCJK_udbline_sep_tl ,
6680   underdblline / format     .tl_set:N = \l__xeCJK_udbline_format_tl ,
6681   underdblline / textformat .tl_set:N = \l__xeCJK_udbline_text_format_tl ,
6682   underdblline / gap        .tl_set:N = \l__xeCJK_udbline_gap_tl ,
6683   underwave / skip          .bool_set:N = \l__xeCJK_uwave_skip_bool ,
6684   underwave / hidden        .bool_set:N = \l__xeCJK_uwave_hidden_bool ,
6685   underwave / subtract      .bool_set:N = \l__xeCJK_uwave_subtract_bool ,
6686   underwave / symbol        .tl_set:N = \l__xeCJK_uwave_symbol_tl ,
6687   underwave / depth         .tl_set:N = \l__xeCJK_uwave_depth_tl ,
6688   underwave / sep           .tl_set:N = \l__xeCJK_uwave_sep_tl ,
6689   underwave / format        .tl_set:N = \l__xeCJK_uwave_format_tl ,
6690   underwave / textformat    .tl_set:N = \l__xeCJK_uwave_text_format_tl ,
6691   sout / skip               .bool_set:N = \l__xeCJK_sout_skip_bool ,
6692   sout / hidden             .bool_set:N = \l__xeCJK_sout_hidden_bool ,
6693   sout / subtract           .bool_set:N = \l__xeCJK_sout_subtract_bool ,
6694   sout / thickness          .tl_set:N = \l__xeCJK_sout_thickness_tl ,
6695   sout / height             .tl_set:N = \l__xeCJK_sout_height_tl ,
6696   sout / format            .tl_set:N = \l__xeCJK_sout_format_tl ,
6697   sout / textformat         .tl_set:N = \l__xeCJK_sout_text_format_tl ,
6698   xout / skip               .bool_set:N = \l__xeCJK_xout_skip_bool ,
6699   xout / hidden             .bool_set:N = \l__xeCJK_xout_hidden_bool ,
6700   xout / subtract           .bool_set:N = \l__xeCJK_xout_subtract_bool ,
6701   xout / format            .tl_set:N = \l__xeCJK_xout_format_tl ,
6702   xout / textformat         .tl_set:N = \l__xeCJK_xout_text_format_tl ,
6703   ulem / skip              .bool_set:N = \l__xeCJK_ulem_skip_bool ,
6704   ulem / hidden            .bool_set:N = \l__xeCJK_ulem_hidden_bool ,
6705   ulem / subtract          .bool_set:N = \l__xeCJK_ulem_subtract_bool ,
6706   ulem / sep               .tl_set:N = \l__xeCJK_ulem_sep_tl ,
6707   ulem / boxdepth          .tl_set:N = \l__xeCJK_ulem_boxdepth_tl ,
6708   ulem / textformat        .tl_set:N = \l__xeCJK_ulem_text_format_tl
6709 }
6710 \clist_map_inline:nn
6711 { underdot , underline , underdblline , underwave , sout , xout , ulem }
6712 {
6713   \keys_define:nn { xeCJK / options }
6714   { #1 .meta:nn = { xeCJK / options / #1 } { ##1 } }
6715 }
6716 \keys_set:nn { xeCJK / options }
6717 {
6718   underdot / symbol      = \normalfont . ,
6719   underdot / depth       = 0.20 em ,
6720   underdot / sep         = 0.04 em ,
6721   symbol / sep           = \c_zero_dim ,
6722   underline / skip       = true ,
6723   underline / thickness  = \ULthickness ,
6724   underline / depth      = 0.20 em ,
6725   underline / sep        = 0.07 em ,
6726   underdblline / skip    = true ,
6727   underdblline / thickness = \ULthickness ,
6728   underdblline / depth   = 0.20 em ,
6729   underdblline / sep     = 0.17 em ,
6730   underdblline / gap     = 1.1 pt ,
6731   underwave / skip       = true ,
6732   underwave / symbol     = \sixty \tex_char:D 58 \exp_stop_f: ,
6733   underwave / depth      = 0.20 em ,
6734   underwave / sep        = 0.00 em ,
6735   sout / skip            = true ,
6736   sout / thickness       = \ULthickness ,
6737   sout / height          = 0.35 em ,
6738   xout / skip            = true
6739 }
\CJKunderanysymbol 6740 \NewDocumentCommand \CJKunderanysymbol { o m m m }
6741 {
6742   \xeCJK_under_symbol:nnnnnn { symbol } { symbol } {#1} {#2} {#3} {#4}
6743 }

```

`\CJKunderdot` `\CJKunderdot` 是 `\CJKunderanysymbol` 的特殊情况。`CJKntef` 原来使用的是数学符号 `\cdot`，这里改成更合适的 `.`。

```
6744 \NewDocumentCommand \CJKunderdot { o m }
6745 {
6746   \xeCJK_under_symbol:nnnnnn { underdot } { udot }
6747   {#1}
6748   { \l__xeCJK_udot_depth_tl }
6749   { \l__xeCJK_udot_format_tl \l__xeCJK_udot_symbol_tl }
6750   {#2}
6751 }
```

`\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。

```
6752 \cs_new_protected:Npn \xeCJK_under_symbol:nnnnnn
6753 {
6754   \xeCJK_if_ulem_patch:TF
6755   { \__xeCJK_under_symbol_auxi:nnnnnn }
6756   { \__xeCJK_under_symbol_auxii:nnnnnn }
6757 }
6758 \cs_new_protected:Npn \__xeCJK_under_symbol_auxi:nnnnnn #1#2#3#4#5#6
6759 {
6760   \xeCJK_ulem_right: \UL@stop
6761   \group_begin:
6762     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
6763     \use:c { l__xeCJK_#2_text_format_tl }
6764     \UL@start \xeCJK_ulem_right_node:
6765     #6
6766     \xeCJK_ulem_right: \UL@stop
6767   \group_end:
6768   \UL@start \xeCJK_ulem_right_node:
6769 }
6770 \cs_new_protected:Npn \__xeCJK_under_symbol_auxii:nnnnnn #1#2#3#4#5#6
6771 {
6772   \mode_leave_vertical:
6773   \group_begin:
6774     \xeCJK_under_symbol_initial:nnnnn {#1} {#2} {#3} {#4} {#5}
6775     \__xeCJK_under_symbol_text_format:c { l__xeCJK_#2_text_format_tl }
6776     #6
6777     \xeCJK_ulem_right:
6778   \group_end:
6779   \xeCJK_ulem_right_node:
6780   \bool_gset_true:N \g__xeCJK_glue_check_pending_bool
6781 }
6782 \cs_new_protected:Npn \xeCJK_under_symbol_initial:nnnnn #1#2#3#4#5
6783 {
6784   \tl_if_novalue:nF {#3}
6785   { \keys_set:nn { xeCJK / options / #1 } {#3} }
6786   \xeCJK_fntef_sbox:n {#5}
6787   \bool_if:NTF \l__xeCJK_fntef_bool
6788   { \xeCJK_make_under_symbol:n { \l__xeCJK_fntef_dim } }
6789   {
6790     \bool_set_true:N \l__xeCJK_fntef_bool
6791     \xeCJK_make_under_symbol:n {#4}
6792   }
6793   \tl_if_empty:cF { l__xeCJK_#2_boxdepth_tl }
6794   {
6795     \box_set_dp:Nn \l__xeCJK_under_symbol_box
6796     { \use:c { l__xeCJK_#2_boxdepth_tl } }
6797   }
6798   \dim_set:Nn \l__xeCJK_fntef_dim
6799   { \use:c { l__xeCJK_#2_sep_tl } + \box_dp:N \l__xeCJK_under_symbol_box }
6800   \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N
6801   \__xeCJK_restore_shipout_CJKsymbol:
6802 }
6803 \cs_new_protected:Npn \__xeCJK_under_symbol_text_format:N #1
6804 {
6805   \tl_if_empty:NF #1
```

```

6806      { \xeCJK_ulem_right: #1 \xeCJK_ulem_right_node: }
6807    }
6808    \cs_generate_variant:Nn \__xeCJK_under_symbol_text_format:N { c }
6809    \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。

```

6810    \cs_new_protected:Npn \xeCJK_make_under_symbol:n #1
6811    {
6812      \hbox_set:Nn \l__xeCJK_under_symbol_box
6813      {
6814        \box_move_down:nn { #1 + \box_ht:N \l__xeCJK_fntef_box }
6815        {
6816          \hbox_to_zero:n
6817          {
6818            \xeCJK_select_font:
6819            \tex_kern:D \tex_fontcharwd:D \tex_font:D "4E00 \exp_stop_f:
6820            \tex_hss:D \box_use:N \l__xeCJK_fntef_box \tex_hss:D
6821          }
6822        }
6823      }
6824    }

```

__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。

```

6825    \cs_new_protected:Npn \__xeCJK_restore_shipout_CJKsymbol:
6826    {
6827      \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
6828      { \xeCJK_swap_cs:NN \CJKsymbol \__xeCJK_under_CJKsymbol:N }
6829      \__xeCJK_restore_shipout_fntef:
6830      \xeCJK_cs_clear:N \__xeCJK_restore_shipout_CJKsymbol:
6831    }
6832    \cs_new_protected:Npn \__xeCJK_restore_shipout_fntef:
6833    {
6834      \tl_put_right:Nn \l__xeCJK_fntef_shipout_tl
6835      {
6836        \bool_set_false:N \l__xeCJK_fntef_bool
6837        \dim_zero:N \l__xeCJK_fntef_dim
6838      }
6839      \xeCJK_cs_clear:N \__xeCJK_restore_shipout_fntef:
6840    }
6841    \tl_new:N \l__xeCJK_fntef_shipout_tl
6842    \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。

```

6843    \cs_new_protected:Npn \__xeCJK_under_CJKsymbol:N
6844    {
6845      \box_use:N \l__xeCJK_under_symbol_box
6846      \xeCJK_no_break: \__xeCJK_under_CJKsymbol:N
6847    }

```

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。

```

6848    \NewDocumentEnvironment { CJKfilltwosides } { 0 { c } m }
6849    {
6850      \use:e { \exp_not:N \minipage [#1] { \dim_eval:n {#2} } }
6851      \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
6852    }
6853    {
6854      \endminipage
6855      \ignorespacesafterend
6856    }
6857    \NewDocumentEnvironment { CJKfilltwosides* } { 0 { c } m +b }
6858    {

```

```

6859 \mode_leave_vertical:
6860 \cs_set_eq:NN \CJKglue \xeCJK_fntef_hfilll:
6861 \tl_set:Nn \arraystretch { 1 }
6862 \cs_if_free:NF \extrarowheight
6863 { \cs_set_eq:NN \extrarowheight \c_zero_dim }
6864 \use:e { \_xeCJK_fill_two_sides:nnn {#1} { \dim_eval:n {#2} } } {#3}
6865 }
6866 { \ignorespacesafterend }
6867 \cs_new_protected:Npn \_xeCJK_fill_two_sides:nnn #1#2#3
6868 {
6869   \dim_compare:nNnTF {#2} > \c_zero_dim
6870   {
6871     \hbox_set:Nn \l__xeCJK_tmp_box
6872     { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
6873     \dim_compare:nNnTF {#2} > { \box_wd:N \l__xeCJK_tmp_box }
6874     { \tabular [#1] { @ { } p {#2} @ { } } #3 \endtabular }
6875     { \box_use:N \l__xeCJK_tmp_box }
6876   }
6877   { \tabular [#1] { @ { } c @ { } } #3 \endtabular }
6878 }

```

\xeCJK_fntef_hfilll: **colortbl** 将表格 c 列用于填充的 \hfil 改为了更高阶的 fill, 影响到了 CJKfillltwosides*。因此, 我们也要用高阶的 filll。

```

6879 \cs_new_protected:Npn \xeCJK_fntef_hfilll:
6880 { \skip_horizontal:N \c__xeCJK_filll_skip }
6881 \skip_const:Nn \c__xeCJK_filll_skip { \c_zero_dim plus 1 filll }
6882 </fntef>

```

5.20 xeCJK-listings

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

```

6883 <*listings>
6884 \DeclareOption* { \PassOptionsToPackage { \CurrentOption } { xeCJK } }
6885 \ProcessOptions \scan_stop:
6886 \RequirePackage { xeCJK }
6887 \RequirePackage { listings }
6888 \lst@AddToHook { Init } { \_xeCJK_listings_initial_hook: }
6889 \lst@AddToHook { SelectCharTable } { \_xeCJK_listings_toks_hook: }
6890 \lst@AddToHook { OutputBox }
6891 {
6892   \tl_set_eq:NN \l_xeCJK_punct_style_tl \c__xeCJK_punct_style_plain_tl
6893   \l__xeCJK_restore_listings_toks_tl
6894   \_xeCJK_listings_output_CM:
6895 }
6896 \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。

```

6897 \cs_new_protected:Npn \_xeCJK_listings_initial_hook:
6898 {
6899   \tex_noindent:D
6900   \bool_gset_false:N \g__xeCJK_listings_CM_bool
6901   \tl_put_left:Nn \lst@numberstyle { \l__xeCJK_restore_listings_toks_tl }
6902   \xeCJK_add_to_shipout:n { \l__xeCJK_restore_listings_toks_tl }
6903   \lst@ifbreaklines
6904     \cs_set_eq:NN \_xeCJK_listings_CJK_toks_hook: \_xeCJK_listings_breaklines_toks:
6905     \tl_if_empty:NF \lst@prebreak

```

```

6906      { \tl_put_left:Nn \lst@prebreak { \l__xeCJK_restore_listings_toks_tl } }
6907      \tl_if_empty:NF \lst@postbreak
6908      { \tl_put_left:Nn \lst@postbreak { \l__xeCJK_restore_listings_toks_tl } }
6909      \fi:
6910      \int_set:Nn \l__xeCJK_listings_max_char_int
6911      { \lst@ifec 255 \else: 127 \fi: }
6912  }
6913  \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。

```

6914  \cs_new_protected:Npn \__xeCJK_listings_toks_hook:
6915  {
6916      \tl_clear:N \l__xeCJK_restore_listings_toks_tl
6917      \seq_map_function:NN
6918      \g__xeCJK_class_seq \__xeCJK_backup_inter_class_toks:n
6919      \seq_map_inline:Nn \g__xeCJK_non_CJK_class_seq
6920      {
6921          \str_if_eq:nnF { ##1 } { Boundary }
6922          {
6923              \xeCJK_inter_class_toks:nnn { Boundary } { ##1 }
6924              { \__xeCJK_listings_process_Default:nN { ##1 } }
6925          }
6926      }
6927      \xeCJK_inter_class_toks:nnn { Boundary } { CM }
6928      { \__xeCJK_listings_process_CM:nN { 0 } }
6929      \__xeCJK_listings_CJK_toks_hook:
6930  }

```

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

```

6931  \cs_new_protected:Npn \__xeCJK_backup_inter_class_toks:n #1
6932  {
6933      \tl_set:Ne \l__xeCJK_tmp_tl
6934      { \xeCJK_get_inter_class_toks:nn { Boundary } {#1} }
6935      \tl_put_right:Ne \l__xeCJK_restore_listings_toks_tl
6936      {
6937          \xeCJK_inter_class_toks:nnn { Boundary } {#1}
6938          {
6939              \tl_if_empty:NTF \l__xeCJK_tmp_tl
6940              { \exp_not:N \prg_do_nothing: }
6941              { \exp_not:o \l__xeCJK_tmp_tl }
6942          }
6943      }
6944  }
6945  \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:
6946  \cs_new_protected:Npn \__xeCJK_listings_CJK_toks_hook:
6947  {
6948      \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
6949      { \__xeCJK_listings_process_CJK:nN { 2 } }
6950      \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
6951      { \__xeCJK_listings_process_CJK:nN { 2 } }
6952      \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
6953      { \__xeCJK_listings_process_CJK:nN { 2 } }
6954      \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
6955      { \__xeCJK_listings_process_CJK:nN { 2 } }
6956      \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
6957      { \__xeCJK_listings_process_CJK:nN { 2 } }
6958      \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
6959      { \__xeCJK_listings_process_CM:nN { 0 } }
6960      \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
6961      { \__xeCJK_listings_process_CM:nN { 0 } }
6962      \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
6963      { \__xeCJK_listings_process_CJK:nN { 2 } }

```

```

6964 \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
6965 {
6966   \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
6967   { \__xeCJK_listings_process_CJK:nN { 2 } }
6968 }
6969 }
6970 \cs_new_protected:Npn \__xeCJK_listings_breaklines_toks:
6971 {
6972   \xeCJK_inter_class_toks:nnn { Boundary } { CJK }
6973   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
6974   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamo }
6975   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
6976   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoL }
6977   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
6978   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoV }
6979   { \__xeCJK_listings_process_CM:nN { 0 } }
6980   \xeCJK_inter_class_toks:nnn { Boundary } { HangulJamoT }
6981   { \__xeCJK_listings_process_CM:nN { 0 } }
6982   \xeCJK_inter_class_toks:nnn { Boundary } { CJStarter }
6983   { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
6984   \xeCJK_inter_class_toks:nnn { Boundary } { FullLeft }
6985   { \__xeCJK_listings_process_FullLeft:nN { 2 } }
6986   \xeCJK_inter_class_toks:nnn { Boundary } { FullRight }
6987   { \__xeCJK_listings_process_FullRight:nN { 2 } }
6988   \seq_map_inline:Nn \g__xeCJK_CJK_sub_class_seq
6989   {
6990     \xeCJK_inter_class_toks:nnn { Boundary } { CJK/##1 }
6991     { \__xeCJK_listings_process_breaklines_CJK:nN { 2 } }
6992   }
6993 }

```

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

```

6994 \cs_new_protected:Npn \__xeCJK_listings_process_Default:nN #1#2
6995 {
6996   \int_compare:nNnTF
6997   { \xeCJK_token_value_charcode:N #2 } > \l__xeCJK_listings_max_char_int
6998   {
6999     \token_if_letter:NTF #2
7000     { \lst@ProcessLetter #2 }
7001     { \lst@ProcessOther #2 }
7002   }
7003   { \__xeCJK_listings_output_Default:nN {#1} #2 }
7004 }

```

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:。

```

7005 \cs_new_protected:Npn \__xeCJK_listings_output_Default:nN #1#2
7006 {
7007   \group_begin:
7008   \xeCJK_clear_inter_class_toks:nn { Boundary } {#1}
7009   \xeCJK_inter_class_toks:nnn {#1} { Boundary } { \group_end: }
7010   #2
7011   \scan_stop:
7012 }

```

__xeCJK_listings_process_CJK:nN 对 CJK 字符类的处理。

```

7013 \cs_new_protected:Npn \__xeCJK_listings_process_CJK:nN #1#2
7014 {
7015   \token_if_letter:NTF #2
7016   { \__xeCJK_listings_process_letter:nN {#1} #2 }
7017   { \__xeCJK_listings_process_other:nN {#1} #2 }
7018 }

```

`\_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。

```
7019 \cs_new_protected:Npn \_xeCJK_listings_append:nN #1#2
7020 {
7021   \int_add:Nn \lst@length { #1 - 1 }
7022   \lst@Append #2
7023 }
```

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

`\_xeCJK_listings_process_other:nN`

```
7024 \cs_new_protected:Npn \_xeCJK_listings_process_letter:nN
7025 {
7026   \lst@whitespacefalse
7027   \bool_if:NTF \l__xeCJK_listings_letter_bool
7028     { \lst@lettertrue }
7029     {
7030       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
7031       \bool_set_true:N \l__xeCJK_listings_letter_bool
7032     }
7033   \_xeCJK_listings_append:nN
7034 }
7035 \bool_new:N \l__xeCJK_listings_letter_bool
7036 \cs_new_protected:Npn \_xeCJK_listings_process_other:nN #1#2
7037 {
7038   \lst@whitespacefalse
7039   \bool_if:NTF \l__xeCJK_listings_letter_bool
7040     {
7041       \lst@Output \lst@letterfalse
7042       \bool_set_false:N \l__xeCJK_listings_letter_bool
7043     }
7044     { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
7045   \cs_set_eq:NN \lst@lastother #2
7046   \_xeCJK_listings_append:nN {#1} #2
7047 }
```

`\_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`

```
7048 \cs_new_protected:Npn \_xeCJK_listings_process_breaklines_CJK:nN
7049 {
7050   \lst@whitespacefalse
7051   \bool_if:NTF \l__xeCJK_listings_letter_bool
7052     {
7053       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2 { \lst@Output }
7054       \lst@lettertrue
7055     }
7056     {
7057       \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
7058       \bool_set_true:N \l__xeCJK_listings_letter_bool
7059     }
7060   \int_set_eq:NN \l__xeCJK_listings_flag_int \c_one_int
7061   \_xeCJK_listings_append:nN
7062 }
7063 \cs_new_protected:Npn \_xeCJK_listings_process_FullLeft:nN #1#2
7064 {
7065   \lst@whitespacefalse
7066   \bool_if:NTF \l__xeCJK_listings_letter_bool
7067     {
7068       \int_compare:nNnF \l__xeCJK_listings_flag_int = 2
7069         {
7070           \int_compare:nNnTF \l__xeCJK_listings_flag_int = 3
7071             { \bool_if:NT \l__xeCJK_punct_breakable_bool { \lst@Output } }
7072             { \lst@Output }
7073         }
7074       \lst@lettertrue
```

```

7075     }
7076     {
7077         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
7078         \bool_set_true:N \l__xeCJK_listings_letter_bool
7079     }
7080     \int_set:Nn \l__xeCJK_listings_flag_int { 2 }
7081     \__xeCJK_listings_append:nN {#1} #2
7082 }
7083 \cs_new_protected:Npn \__xeCJK_listings_process_FullRight:nN #1#2
7084 {
7085     \lst@whitespacefalse
7086     \bool_if:NTF \l__xeCJK_listings_letter_bool
7087     {
7088         \int_compare:nNt \l__xeCJK_listings_flag_int < 2
7089         { \__xeCJK_punct_if_long:NT #2 { \lst@Output } }
7090         \lst@lettertrue
7091     }
7092     {
7093         \lst@ifletter \lst@Output \else: \lst@OutputOther \lst@lettertrue \fi:
7094         \bool_set_true:N \l__xeCJK_listings_letter_bool
7095     }
7096     \int_set:Nn \l__xeCJK_listings_flag_int { 3 }
7097     \__xeCJK_listings_append:nN {#1} #2
7098 }
7099 \int_new:N \l__xeCJK_listings_flag_int

\lst@AppendLetter 7100 \cs_set_protected:Npn \lst@AppendLetter
\lst@AppendOther 7101 {
7102     \bool_if:NTF \l__xeCJK_listings_letter_bool
7103     {
7104         \lst@Output \lst@lettertrue
7105         \bool_set_false:N \l__xeCJK_listings_letter_bool
7106     }
7107     { \reverse_if:N \lst@ifletter \lst@OutputOther \lst@lettertrue \fi: }
7108     \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
7109     \lst@Append
7110 }
7111 \cs_set_protected:Npn \lst@AppendOther
7112 {
7113     \bool_if:NTF \l__xeCJK_listings_letter_bool
7114     {
7115         \lst@Output \lst@letterfalse
7116         \bool_set_false:N \l__xeCJK_listings_letter_bool
7117     }
7118     { \lst@ifletter \lst@Output \lst@letterfalse \fi: }
7119     \lst@ifbreaklines \int_zero:N \l__xeCJK_listings_flag_int \fi:
7120     \tex_futurelet:D \lst@lastother \lst@Append
7121 }

```

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

```

7122 \cs_new_protected:Npn \__xeCJK_listings_process_CM:nN
7123 {
7124     \reverse_if:N \lst@ifflexible
7125     \bool_gset_true:N \g__xeCJK_listings_CM_bool
7126     \fi:
7127     \__xeCJK_listings_process_letter:nN
7128 }

```

__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 X_YTeX 将基本字和组合标识正确的组合起来。

```

7129 \cs_new_protected:Npn \__xeCJK_listings_output_CM:
7130 {
7131     \reverse_if:N \lst@ifflexible
7132     \bool_if:NT \g__xeCJK_listings_CM_bool
7133     {
7134         \bool_gset_false:N \g__xeCJK_listings_CM_bool

```

```

7135      \xeCJK_cs_clear:N \lst@FillOutputBox
7136      \cs_set_eq:NN \CJKglue \tex_hss:D
7137    }
7138    \fi:
7139  }
7140  \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。

```

7141 \cs_new_protected:Npn \__xeCJK_listings_peek_active_loop:TF #1#2#3
7142 {
7143   \token_if_active:NTF #3
7144   { #1#3 }
7145   {
7146     \token_if_cs:NTF #3
7147     { #2#3 }
7148     {
7149       \int_compare:nNnTF {`#3} > { \l__xeCJK_listings_max_char_int }
7150       { \__xeCJK_listings_peek_active_loop:TF { #1#3 } { #2#3 } }
7151       { #2#3 }
7152     }
7153   }
7154 }
7155 \cs_set_eq:NN \lst@ifnextcharactive \__xeCJK_listings_peek_active_loop:TF

```

_xeCJK_listings_rescan:Nn 当 \lstinline 被使用在参数中时, listings 会使用一个循环逐个将 \lstinline 参数中的字符设置为活动字符。我们可以通过 \tl_set_rescan:Nnn 来完成这里的 \catcode 转换, 避免将 \charcode 超过 255 的字符都设置为活动字符。

当 \lstinline 出现在宏参数中时, TeX 参数传递将 catcode 6 的 parameter token (通常是 #) 双写为成对的相邻 token。 \tl_set_rescan:Nnn 底层的 \scantokens 在字符串化阶段会再次将 catcode 6 token 双写, 导致输出中数量翻倍。修复方法是在 rescan 前逐 token 扫描 \l__xeCJK_tmp_tl, 将每个 catcode 6 的 token 替换为保留原字符码的 active token (catcode 13), 这是 listings 期望的字符类型。最初的实现用 \regex_replace_all:nnN 把所有 catcode 6 token 硬替换成 U+0023, 对一般 # 没问题, 但当用户用 \catcode 把别的字符(如 &)改为 parameter 时, 原字符码会被丢失, 导致输出错字(#879)。

```

7156 \cs_new_protected:Npn \__xeCJK_listings_rescan:Nn #1#2
7157 {
7158   \__xeCJK_listings_set_escape:
7159   \tl_set:Nn \l__xeCJK_tmp_tl {#2}
7160   \__xeCJK_listings_escape_backslash:
7161   \tl_clear:N \l__xeCJK_tmpb_tl
7162   \tl_map_inline:Nn \l__xeCJK_tmp_tl
7163   {
7164     \token_if_parameter:NTF ##1
7165     {
7166       \tl_put_right:Ne \l__xeCJK_tmpb_tl
7167       {
7168         \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:N
7169         \char_generate:nn { \int_value:w `##1 } { 13 }
7170       }
7171     }
7172     { \tl_put_right:Nn \l__xeCJK_tmpb_tl { ##1 } }
7173   }
7174   \tl_set_eq:NN \l__xeCJK_tmp_tl \l__xeCJK_tmpb_tl
7175   \tl_set_rescan:Nno #1
7176   { \char_set_catcode_active:n { 35 } }
7177   { \l__xeCJK_tmp_tl }
7178 }
7179 \cs_new_protected:Npn \__xeCJK_listings_inside_convert:nw #1 ~ \@empty
7180 {
7181   \__xeCJK_listings_rescan:Nn \l__xeCJK_tmp_tl {#1}

```

```

7182 \tl_put_right:No \lst@arg { \l__xeCJK_tmp_tl }
7183 }
7184 \cs_set_eq:NN \lst@InsideConvert@ \__xeCJK_listings_inside_convert:nw
7185 \cs_new_protected:Npn \__xeCJK_listings_inline_group:w
7186 {
7187   \exp_after:wN \__xeCJK_listings_inline_group:n
7188   \exp_after:wN { \if_int_compare:w ` } = \c_zero_int \fi:
7189 }
7190 \cs_set_eq:NN \lst@InlineGJ \__xeCJK_listings_inline_group:w
7191 \cs_new_protected:Npn \__xeCJK_listings_inline_group:n #1
7192 {
7193   \__xeCJK_listings_rescan:Nn \lst@arg {#1}
7194   \lst@InlineGJEnd
7195 }

```

`\__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。

```

7196 \group_begin:
7197 \cs_set:Npn \__xeCJK_tmp:w #1
7198 {
7199   \group_end:
7200   \cs_new_protected:Npn \__xeCJK_listings_set_escape:
7201     { \xeCJK_swap_cs:NN #1 \__xeCJK_listings_escape:N }
7202   \cs_new_protected:Npn \__xeCJK_listings_escape:N ##1
7203     { \cs_if_eq:NNTF #1 ##1 { \__xeCJK_listings_escape:N } {##1} }
7204 }
7205 \use:n
7206 {
7207   \char_set_catcode_active:N \
7208   \__xeCJK_tmp:w
7209 }
7210 { \ }

```

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

```

7211 \cs_new_protected:Npe \__xeCJK_listings_escape_backslash:
7212 {
7213   \tl_replace_all:Nnn \exp_not:N \l__xeCJK_tmp_tl
7214     { \c_backslash_str }
7215     { \c_backslash_str \c_backslash_str }
7216 }
7217 </listings>
7218 <@@=xunadd>

```

5.21 xunicode-addon

```
7219 <*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。

```

7220 \bool_lazy_or:nnF
7221 { \sys_if_engine_xetex_p: }
7222 { \sys_if_engine luatex_p: }
7223 {
7224   \msg_new:nnnn { xunicode-addon } { xetex-luatex }
7225   { This~package~requires~either~XeTeX~or~LuaTeX~to~function.}
7226   {

```

```

7227         You~must~change~your~typesetting~engine~to,~e.g.,\\
7228         "xelatex"~or~"lualatex"~instead~of~plain~"latex"~or~"pdflatex".
7229     }
7230     \msg_critical:nn { xunicode-addon } { xetex-luatex }
7231 }
7232 \RequirePackage { xparse }

```

宏包选项是编码的名字。

```

7233 \clist_new:N \g__xunadd_encname_clist
7234 \tl_if_exist:NT \UTFencname
7235 { \clist_gput_right:Ne \g__xunadd_encname_clist { \UTFencname } }
7236 \DeclareOption*
7237 { \clist_gput_right:No \g__xunadd_encname_clist \CurrentOption }
7238 \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。

```

7239 \@ifpackageloaded { xunicode } { }
7240 {
7241     \clist_get:NNF \g__xunadd_encname_clist \UTFencname
7242     {
7243         \cs_if_exist:NTF \UnicodeEncodingName
7244         { \tl_set:Ne \UTFencname { \UnicodeEncodingName } }
7245         {
7246             \sys_if_engine_xetex:TF
7247             { \tl_set:Nn \UTFencname { EU1 } }
7248             { \tl_set:Nn \UTFencname { EU2 } }
7249         }
7250         \clist_gset_eq:NN \g__xunadd_encname_clist \UTFencname
7251     }
7252     \sys_if_engine_xetex:TF
7253     { \RequirePackage { xunicode } }
7254     {
7255         \cs_set_eq:NN \__xunadd_tmp:w \XeTeXpicfile
7256         \cs_set_eq:NN \XeTeXpicfile \prg_do_nothing:
7257         \RequirePackage { xunicode }
7258         \cs_set_eq:NN \XeTeXpicfile \__xunadd_tmp:w
7259     }
7260 }
7261 \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。

```

7262 \RenewDocumentCommand \ReloadXunicode { m }
7263 {
7264     \clist_set:Ne \l__xunadd_encname_clist {#1}
7265     \__xunadd_reload:N \l__xunadd_encname_clist
7266 }
7267 \cs_new_protected:Npn \__xunadd_reload:N #1
7268 {
7269     \cs_set_eq:NN \__xunadd_tmp:w \iftipaonetoken
7270     \cs_set_eq:NN \iftipaonetoken \scan_stop:
7271     \use:e
7272     {
7273         \ExplSyntaxOff
7274         \char_set_catcode_letter:n { 64 }
7275         \exp_not:N \clist_map_function:NN \exp_not:N #1 \__xunadd_reload_aux:n
7276         \bool_if:NTF \l__kernel_expl_bool
7277         { \ExplSyntaxOn }
7278         { \ExplSyntaxOff }
7279         \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
7280     }
7281     \cs_set_eq:NN \iftipaonetoken \__xunadd_tmp:w
7282 }
7283 \cs_new_protected:Npn \__xunadd_reload_aux:n #1

```

```

7284 {
7285   \cs_if_exist:cTF { T@ #1 }
7286   {
7287     \tl_set:Nn \UTFencname {#1}
7288     \clist_if_in:NnF \g__xunadd_encname_clist {#1}
7289     { \clist_gput_right:Nn \g__xunadd_encname_clist {#1} }
7290     \file_input:n { xunicode.sty }
7291     \file_input:n { xunicode-extra.def }
7292   }
7293   { \msg_error:nnn { xunicode-addon } { encoding-unknown } {#1} }
7294 }
7295 \clist_new:N \l__xunadd_encname_clist
7296 \msg_new:nnnn { xunicode-addon } { encoding-unknown }
7297 { Encoding~scheme~"#1"~unknown. }
7298 {
7299   You~may~use \\\
7300   \token_to_str:N \usepackage [ #1 , \encodingdefault ] \{fontenc\} \\\
7301   before~xunicode-addon~or~xunicode.
7302 }

```

\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。

```

7303 \RenewDocumentCommand \DeclareUTFmathsymbols { m }
7304 {
7305   \bool_if:NT \l__xunadd_math_as_UTF_text_bool
7306   {
7307     \seq_map_inline:Nn \l__xunadd_math_as_UTF_text_seq
7308     { \__xunadd_declare_math_as_UTF_text:n {##1} }
7309     \bool_set_false:N \l__xunadd_math_as_UTF_text_bool
7310   }
7311 }
7312 \seq_new:N \l__xunadd_math_as_UTF_text_seq
7313 \seq_set_from_clist:Nn \l__xunadd_math_as_UTF_text_seq
7314 { hbar , Finv , aleph , beth , gimel , dalet , Game }
7315 \bool_new:N \l__xunadd_math_as_UTF_text_bool
7316 \RenewDocumentCommand \UseMathAsText { }
7317 {
7318   \math@s@text@true
7319   \bool_set_true:N \l__xunadd_math_as_UTF_text_bool
7320 }
7321 \@onlypreamble \UseMathAsText
7322 \cs_new_protected:Npn \__xunadd_declare_math_as_UTF_text:n #1
7323 {
7324   \cs_if_exist:cTF {#1}
7325   {
7326     \cs_new_eq:cc { keepmathUTF #1 } {#1}
7327     \cs_gset_protected:cpe {#1}
7328     {
7329       \exp_not:N \mode_if_math:TF
7330       { \exp_not:c { keepmathUTF #1 } }
7331       { \exp_not:c { text #1 } }
7332     }
7333     \tl_put_right:Ne \l__xunadd_hyperref_hook_tl
7334     { \cs_set_eq:NN \exp_not:c {#1} \exp_not:c { text #1 } }
7335   }
7336   { \cs_new:cpe {#1} { \exp_not:c { text #1 } } }
7337 }
7338 \tl_new:N \l__xunadd_hyperref_hook_tl
7339 \AtBeginDocument
7340 {
7341   \cs_if_free:NF \pdfstringdefDisableCommands
7342   { \pdfstringdefDisableCommands { \l__xunadd_hyperref_hook_tl } }
7343 }

```

__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
7344 \prg_new_conditional:Npnn \__xunadd_glyph_if_exist:n #1 { p , T , F , TF }
7345 {

```

```

7346 \tex_iffontchar:D \tex_font:D \tex_numexpr:D #1 \scan_stop:
7347 \prg_return_true: \else: \prg_return_false: \fi:
7348 }

```

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

```

7349 \RenewDocumentCommand \UndeclareUTFcharacter { O { \UTFencname } m m }
7350 {
7351   \__xunadd_if_csname:nTF {#3}
7352     { \UndeclareTextCommand {#3} }
7353     { \exp_args:Nc \UndeclareTextCommand { \tl_to_str:n {#3} } }
7354     {#1}
7355 }

```

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

```

7356 \RenewDocumentCommand \UndeclareUTFcomposite { O { \UTFencname } m m m }
7357 {
7358   \__xunadd_if_csname:nTF {#3}
7359     { \__xunadd_undeclare_composite:Nnnn #3 }
7360     { \exp_args:Nc \__xunadd_undeclare_composite:Nnnn { \tl_to_str:n {#3} } }
7361     {#1} {#4} {#2}
7362 }
7363 \cs_new_protected:Npn \__xunadd_undeclare_composite:Nnnn #1#2#3#4
7364 { \cs_undefine:c { \__xunadd_composite_cs:Nnn #1 {#2} {#3} } }

```

```

\__xunadd_composite_cs:Nnn 7365 \cs_new:Npe \__xunadd_composite_cs:Nnn #1#2#3
\__xunadd_composite_cs:nnn 7366 { \c_backslash_str #2 \exp_not:N \token_to_str:N #1 - \exp_not:N \tl_to_str:n {#3} }
7367 \cs_new:Npe \__xunadd_composite_cs:nnn #1#2#3
7368 { \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}

7369 \prg_new_conditional:Npnn \__xunadd_if_csname:n #1 { TF }
7370 {
7371   \tl_if_single_token:nTF {#1}
7372   {
7373     \token_if_cs:NTF #1
7374     { \prg_return_true: }
7375     {
7376       \token_if_active:NTF #1
7377       { \prg_return_true: }
7378       { \prg_return_false: }
7379     }
7380   }
7381   { \prg_return_false: }
7382 }

```

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

```

7383 \RenewDocumentCommand \DeclareUTFcharacter { O { \UTFencname } m m }
7384 {
7385   \cs_if_exist_use:cF
7386     { \__xunadd_restore_ \tl_to_str:n {#3} : }
7387     {
7388       \__xunadd_if_csname:nTF {#3}
7389       { \__xunadd_declare_character:Nnn #3 }
7390       { \__xunadd_declare_character:cnn { \tl_to_str:n {#3} } }
7391       {#1} {#2}
7392     }
7393 }

```

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

```

7394 \cs_new_protected:cpn
7395 { \__xunadd_restore_ \tl_to_str:n { \hbar } : }
7396 { \__xunadd_restore_cmd:N \hbar }
7397 \cs_new_protected:cpn

```

```

7398 { __xunadd_restore_ \tl_to_str:n { \nobreakspace } : }
7399 { __xunadd_restore_cmd:N \nobreakspace }
7400 \cs_new_protected:Npn __xunadd_restore_cmd:N #1
7401 { __xunadd_restore_cmd:Ne #1 { ? - \token_to_str:N #1 } }
7402 \cs_new_protected:Npn __xunadd_restore_cmd:Nn #1#2
7403 {
7404   \cs_if_free:cF {#2}
7405   { __xunadd_restore_cmd:Nc #1 {#2} }
7406 }
7407 \cs_new_protected:Npn __xunadd_restore_cmd:NN #1#2
7408 {
7409   \cs_gset_eq:NN #1 #2
7410   \cs_undefine:N #2
7411 }
7412 \cs_generate_variant:Nn __xunadd_restore_cmd:Nn { Ne }
7413 \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。

```

7414 \cs_new_protected:Npn __xunadd_declare_character:Nnn #1#2#3
7415 {
7416   __xunadd_provide_text_command_default:N #1
7417   \exp_after:wN __xunadd_declare_character:NNen
7418   \tex_Uchar:D __xunadd_check_slot:n {#3} \exp_stop_f:
7419   #1 { \token_to_str:N #1 } {#2}
7420 }
7421 \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。

```

7422 \NewDocumentCommand \DeclareUTFSymbol { m O { \UTFencname } m }
7423 { __xunadd_declare_character:Nnn #1 {#2} {#3} }
7424 \NewDocumentCommand \DeclareUTFCommand { m O { \UTFencname } m }
7425 { __xunadd_text_command:Nonn #1 { \token_to_str:N #1 } {#2} {#3} }
7426 \cs_new_protected:Npn __xunadd_text_command:Nnnn #1#2#3#4
7427 { \DeclareTextCommand #1 {#3} { __xunadd_text_command:nn {#2} {#4} } }
7428 \cs_generate_variant:Nn __xunadd_text_command:Nnnn { No }
7429 \cs_new_protected:Npn __xunadd_text_command:nn #1#2
7430 {
7431   __xunadd_begin_hook:nn {#1} {#2}
7432   #2
7433   __xunadd_end_hook:nn {#1} {#2}
7434 }

```

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

```

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

```

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

```

7435 \cs_new_protected:Npn __xunadd_provide_text_command_default:N #1
7436 {
7437   \cs_if_exist:cF { ? \token_to_str:N #1 }
7438   {
7439     \cs_if_free:cF { ? - \token_to_str:N #1 }
7440     {
7441       \exp_args:NNv \ProvideTextCommandDefault #1
7442       { ? - \token_to_str:N #1 }
7443     }

```

```

7444     }
7445 }

```

`\__xunadd_declare_character:Nnnn` 使用编码 #4 下的符号命令 #2 的时候先判断它对应的实际字符 #1 在当前字体中是否存在。如果不存在则转换到 `\DeclareTextSymbolDefault` 中设置的编码或者使用 `\DeclareTextCommandDefault` 中设置的命令。

```

7446 \cs_new_protected:Npn \__xunadd_declare_character:Nnnn #1#2#3#4
7447 { \DeclareTextCommand #2 {#4} { \__xunadd_text_character:nN {#3} {#1} } }
7448 \cs_new_protected:Npn \__xunadd_text_character:nN #1#2
7449 {
7450   \__xunadd_begin_hook:nn {#1} {#2}
7451   \__xunadd_glyph_if_exist:nTF { `#2 }
7452     {#2} { \cs_if_exist_use:cF { ? #1 } {#2} }
7453   \__xunadd_end_hook:nn {#1} {#2}
7454 }
7455 \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。

```

7456 \cs_new:Npn \__xunadd_check_slot:n #1
7457 {
7458   \int_eval:n
7459   {
7460     \tl_if_head_eq_charcode:nNTF {#1} x
7461     { " \use_none:n #1 } {#1}
7462   }
7463 }

```

`\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。

```

7464 \RenewDocumentCommand \DeclareUTFcomposite { 0 { \UTFencname } m m m }
7465 {
7466   \__xunadd_if_csname:nTF {#3}
7467   { \__xunadd_declare_composite:Nnnn #3 }
7468   { \__xunadd_declare_composite:cnnn { \tl_to_str:n {#3} } }
7469   {#1} {#4} {#2}
7470 }

```

`\__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。

```

7471 \cs_new_protected:Npn \__xunadd_declare_composite:Nnnn #1#2#3#4
7472 {
7473   \tex_afterassignment:D \use_none_delimit_by_q_stop:w
7474   \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
7475   { \__xunadd_check_slot:n {#4} }
7476   \q_stop
7477 }
7478 \cs_new_protected:Npn \__xunadd_chardef:Nn #1#2
7479 { \tex_chardef:D #1 = \tex_numexpr:D #2 \scan_stop: }
7480 \cs_generate_variant:Nn \__xunadd_chardef:Nn { c }
7481 \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。

```

7482 \NewDocumentCommand \DeclareUTFCompositeCommand { m 0 { \UTFencname } m m }
7483 { \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。

```

7484 \NewDocumentCommand \DeclareUTFCompositeSymbol { m 0 { \UTFencname } m m }
7485 {

```

```

7486 \__xunadd_chardef:cn { \__xunadd_composite_cs:Nnn #1 {#2} {#3} }
7487 { \__xunadd_check_slot:n {#4} }
7488 }

```

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

```

7489 \NewDocumentCommand \DeclareUTFComposite { m O { \UTFencname } }
7490 { \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。

```

7491 \NewDocumentCommand \DeclareUTFEncodedAccent { m O { \UTFencname } m m }
7492 { \__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。

```

7493 \NewDocumentCommand \DeclareUTFEncodedAccents { m O { \UTFencname } m m }
7494 { \__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。

```

7495 \NewDocumentCommand \DeclareUTFEncodedSymbol { m O { \UTFencname } m m }
7496 { \__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。

```

7497 \NewDocumentCommand \DeclareUTFEncodedCircle { m O { \UTFencname } m m }
7498 { \__xunadd_declare_encoded:NNnnn \__xunadd_combine_circle:nnNNn #1 {#2} {#3} {#4} }

```

\DeclareEncodedCompositeCharacter 7499 \RenewDocumentCommand \DeclareEncodedCompositeCharacter { m m m m }
7500 { \DeclareUTFEncodedSymbol #2 [#1] { "#3 } { "0#4 } }

\DeclareEncodedCompositeAccents 7501 \RenewDocumentCommand \DeclareEncodedCompositeAccents { m m m m }
7502 { \DeclareUTFEncodedAccents #2 [#1] { "#4 } { "#3 } }

\DeclareUTFDoubleEncodedAccent 7503 \NewDocumentCommand \DeclareUTFDoubleEncodedAccent { m O { \UTFencname } m m }
7504 { __xunadd_declare_encoded:NNnnn __xunadd_combine_double_accent:nnNNn #1 {#2} {#3} {#4} }

\DeclareUTFDoubleEncodedSymbol 7505 \NewDocumentCommand \DeclareUTFDoubleEncodedSymbol { m O { \UTFencname } m m }
7506 { __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。

```

7507 \cs_new_protected:Npn \__xunadd_declare_composite:Nnn #1#2#3
7508 { \DeclareTextCommand #1 {#3} { \__xunadd_text_composite:nnn {#2} {#3} } }

```

```

\__xunadd_text_composite:nnn 7509 \cs_new_protected:Npn \__xunadd_text_composite:nnn #1#2#3
7510 {
7511   \__xunadd_begin_hook:nn {#1} {#3}
7512   \cs_if_exist:cTF { \__xunadd_composite_cs:nnn {#1} {#2} {#3} }
7513   {
7514     \__xunadd_text_composite:cnn
7515     { \__xunadd_composite_cs:nnn {#1} {#2} {#3} } {#1} {#3}
7516   }
7517   { \cs_if_exist_use:cTF { ? #1 } { {#3} } {#3} }
7518   \__xunadd_end_hook:nn {#1} {#3}
7519 }
7520 \cs_new_protected:Npn \__xunadd_text_composite:Nnn #1#2#3
7521 {
7522   \token_if_chardef:NTF #1
7523   {
7524     \__xunadd_glyph_if_exist:nTF {#1}
7525     {#1} { \cs_if_exist_use:cTF { ? #2 } { {#3} } {#3} }
7526   }
7527   {#1}
7528 }
7529 \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。

```

7530 \cs_new_protected:Npn \_xunadd_declare_encoded:Nnnnn #1#2#3#4#5
7531 {
7532   \exp_after:wN \_xunadd_declare_encoded:NNNNee
7533   \tex_Uchar:D \_xunadd_check_slot:n {#4} \exp_after:wN \exp_stop_f:
7534   \tex_Uchar:D \_xunadd_check_slot:n {#5} \exp_stop_f:
7535   #1 #2 { \token_to_str:N #2 } {#3}
7536 }
7537 \cs_new_protected:Npn \_xunadd_declare_encoded:NNNNnn #1#2#3#4#5#6
7538 { \DeclareTextCommand #4 {#6} { #3 {#5} {#6} {#1} {#2} } }
7539 \cs_generate_variant:Nn \_xunadd_declare_encoded:Nnnnn { c }
7540 \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。

```

7541 \cs_new_protected:Npn \_xunadd_text_combine:NnnNNn #1#2#3#4#5#6
7542 {
7543   \_xunadd_begin_hook:nn {#2} {#6}
7544   \cs_if_exist:cTF { \_xunadd_composite_cs:nnn {#2} {#3} {#6} }
7545   {
7546     \_xunadd_text_combine:cNnNNn
7547     { \_xunadd_composite_cs:nnn {#2} {#3} {#6} } #1 {#2} {#4} {#5} {#6}
7548   }
7549   { #1 {#6} {#2} {#4} {#5} }
7550   \_xunadd_end_hook:nn {#2} {#6}
7551 }
7552 \cs_new_protected:Npn \_xunadd_text_combine:NNnNNn #1#2#3#4#5#6
7553 {
7554   \token_if_chardef:NTF #1
7555   { \_xunadd_glyph_if_exist:nTF {#1} {#1} { #2 {#6} {#3} {#4} {#5} } }
7556   {#1}
7557 }
7558 \cs_generate_variant:Nn \_xunadd_text_combine:NNnNNn { c }

```

```

\_xunadd_combine_symbol:nnNNn 7559 \cs_new_protected:Npn \_xunadd_combine_symbol:nnNNn
7560 { \_xunadd_text_combine:NnnNNn \_xunadd_add_symbol:nnNN }
7561 \cs_new_protected:Npn \_xunadd_add_symbol:nnNN #1#2#3#4
7562 {
7563   \tl_if_blank:nTF {#1}
7564   {
7565     \_xunadd_glyph_if_exist:nTF { `#4 }
7566     {#4}
7567     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
7568   }
7569   {
7570     \_xunadd_glyph_if_exist:nTF { `#3 }
7571     { #1#3 }
7572     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
7573   }
7574 }

```

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

```

7575 \cs_new_protected:Npn \_xunadd_combine_accent:nnNNn
7576 { \_xunadd_text_combine:NnnNNn \_xunadd_add_accent:nnNN }
7577 \cs_new_protected:Npn \_xunadd_add_accent:nnNN #1#2#3#4
7578 {
7579   \tl_if_blank:nTF {#1}

```

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

```

7580     {
7581       \__xunadd_glyph_if_exist:nTF { `#4 }
7582       {#4}
7583       { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
7584     }
7585     {
7586       \__xunadd_glyph_if_exist:nTF { `#3 }
7587       { #1#3 }
7588       {
7589         \__xunadd_glyph_if_exist:nTF { `#4 }
7590         { \add@accent { `#4 } {#1} }
7591         { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
7592       }
7593     }
7594   }

```

```

\__xunadd_combine_accents:nnNNn 7595 \cs_new_protected:Npn \__xunadd_combine_accents:nnNNn
\__xunadd_add_accents:nnNN 7596 { \__xunadd_text_combine:NnnNNn \__xunadd_add_accents:nnNN }
7597 \cs_new_protected:Npn \__xunadd_add_accents:nnNN #1#2#3#4
7598 {
7599   \tl_if_blank:nTF {#1}
7600   { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
7601   {
7602     \__xunadd_glyph_if_exist:nTF { `#3 }
7603     { \__xunadd_glyph_if_exist:nTF { `#4 } }
7604     { \use_ii:nn }
7605     { #1#3#4 }
7606     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3#4 } }
7607   }
7608 }

```

__xunadd_combine_circle:nnNNn 对圆圈中的数字或者字母适当缩小, 以适合圆圈的大小。只有字体中存在 U+25EF 时, 才使用
 __xunadd_add_circle:nnNN 这里的设置, 否则还是 L^AT_EX 中的设置。
 __xunadd_add_circle:nN

```

7609 \cs_new_protected:Npn \__xunadd_combine_circle:nnNNn
7610 { \__xunadd_text_combine:NnnNNn \__xunadd_add_circle:nnNN }
7611 \cs_new_protected:Npn \__xunadd_add_circle:nnNN #1#2#3#4
7612 {
7613   \tl_if_blank:nTF {#1}
7614   {
7615     \__xunadd_glyph_if_exist:nTF { `#4 }
7616     {#4}
7617     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
7618   }
7619   {
7620     \__xunadd_glyph_if_exist:nTF { `#4 }
7621     { \__xunadd_add_circle:nN {#1} #4 }
7622     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#1} }
7623   }
7624 }
7625 \cs_new_protected:Npn \__xunadd_add_circle:nN #1#2
7626 {
7627   \hcoffin_set:Nn \l__xunadd_tmp_coffin {#1}
7628   \hcoffin_set:Nn \l__xunadd_circle_coffin {#2}
7629   \fp_set:Nn \l__xunadd_circle_scale_fp
7630   {
7631     \dim_to_decimal_in_unit:nn
7632     {
7633       \fp_use:N \l__xunadd_circle_ratio_fp
7634       \coffin_wd:N \l__xunadd_circle_coffin
7635     }
7636     { \coffin_wd:N \l__xunadd_tmp_coffin }
7637   }
7638   \coffin_scale:Nnn \l__xunadd_tmp_coffin
7639   { \l__xunadd_circle_scale_fp } { \l__xunadd_circle_scale_fp }
7640   \coffin_attach:NnnNnnnn
7641   \l__xunadd_circle_coffin { hc } { vc }
7642   \l__xunadd_tmp_coffin { hc } { vc } { \c_zero_dim } { \c_zero_dim }

```

```

7643 \coffin_typeset:Nnnnn \l__xunadd_circle_coffin
7644 { H } { l } { \c_zero_dim } { \c_zero_dim }
7645 }
7646 \fp_new:N \l__xunadd_circle_scale_fp
7647 \coffin_new:N \l__xunadd_tmp_coffin
7648 \coffin_new:N \l__xunadd_circle_coffin

```

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

```

7649 \NewDocumentCommand \settextcircledratio { m }
7650 { \fp_set:Nn \l__xunadd_circle_ratio_fp {#1} }
7651 \fp_new:N \l__xunadd_circle_ratio_fp
7652 \settextcircledratio { 0.7 }

```

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

```

7653 \cs_new_protected:Npn \__xunadd_combine_double_accent:nnNNn
7654 { \__xunadd_text_combine:NnnNNn \__xunadd_add_double_accent:nnNN }
7655 \cs_new_protected:Npn \__xunadd_add_double_accent:nnNN #1#2#3#4
7656 {
7657   \tl_if_blank:nTF {#1}
7658   {
7659     \__xunadd_glyph_if_exist:nTF { `#4 }
7660     {#4}
7661     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
7662   }
7663   {
7664     \__xunadd_glyph_if_exist:nTF { `#3 }
7665     { \__xunadd_add_double_symbol:nN {#1} #3 }
7666     {
7667       \__xunadd_glyph_if_exist:nTF { `#4 }
7668       { \add@accent { `#4 } {#1} }
7669       { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
7670     }
7671   }
7672 }

```

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

```

7673 \cs_new_protected:Npn \__xunadd_combine_double_symbol:nnNNn
7674 { \__xunadd_text_combine:NnnNNn \__xunadd_add_double_symbol:nnNN }
7675 \cs_new_protected:Npn \__xunadd_add_double_symbol:nnNN #1#2#3#4
7676 {
7677   \tl_if_blank:nTF {#1}
7678   {
7679     \__xunadd_glyph_if_exist:nTF { `#4 }
7680     {#4}
7681     { \cs_if_exist_use:cTF { ? #2 } { {#1} } {#4} }
7682   }
7683   {
7684     \__xunadd_glyph_if_exist:nTF { `#3 }
7685     { \__xunadd_add_double_symbol:nN {#1} #3 }
7686     { \cs_if_exist_use:cTF { ? #2 } { {#1} } { #1#3 } }
7687   }
7688 }

```

`\__xunadd_add_double_symbol:nN` 如果参数的第一个记号是字母类、其他符号类或者由 `\chardef` 定义, 则将组合符号放在它的右边, 否则不作处理。

```

7689 \cs_new_protected:Npn \__xunadd_add_double_symbol:nN #1#2
7690 {
7691   \tl_if_head_is_N_type:nTF {#1}
7692   {
7693     \exp_after:wN \exp_after:wN \exp_after:wN
7694     \__xunadd_add_double_symbol_aux:NnN \exp_after:wN \exp_after:wN
7695     \tl_head:w #1 \q_stop \exp_after:wN { \use_none:n #1 } #2
7696   }
7697   { #1#2 }

```

```

7698 }
7699 \cs_new_protected:Npn \__xunadd_add_double_symbol_aux:NnN #1#2#3
7700 {
7701   \bool_lazy_any:nTF
7702   {
7703     { \token_if_letter_p:N #1 }
7704     { \token_if_other_p:N #1 }
7705     { \token_if_chardef_p:N #1 }
7706   }
7707   { #1#3#2 }
7708   { #1#2#3 }
7709 }

```

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

```

7710 \NewDocumentCommand \AtBeginUTFCommand { s O { } +m }
7711 {
7712   \tl_if_blank:nTF {#2}
7713   {
7714     \IfBooleanTF {#1}
7715     { \xunadd_set_begin_hook:n }
7716     { \xunadd_append_begin_hook:n }
7717   }
7718   { \xunadd_set_begin_hook:nn {#2} }
7719   {#3}
7720 }
7721 \NewDocumentCommand \AtEndUTFCommand { s O { } +m }
7722 {
7723   \tl_if_blank:nTF {#2}
7724   {
7725     \IfBooleanTF {#1}
7726     { \xunadd_set_end_hook:n }
7727     { \xunadd_append_end_hook:n }
7728   }
7729   { \xunadd_set_end_hook:nn {#2} }
7730   {#3}
7731 }

```

```

\xunadd_set_begin_hook:n 7732 \cs_new_protected:Npn \xunadd_set_begin_hook:n
\xunadd_set_end_hook:n 7733 { \tl_set:Nn \l__xunadd_begin_hook_tl }
7734 \cs_new_protected:Npn \xunadd_append_begin_hook:n
7735 { \tl_put_right:Nn \l__xunadd_begin_hook_tl }
7736 \cs_new_protected:Npn \xunadd_set_end_hook:n
7737 { \tl_set:Nn \l__xunadd_end_hook_tl }
7738 \cs_new_protected:Npn \xunadd_append_end_hook:n
7739 { \tl_put_right:Nn \l__xunadd_end_hook_tl }
7740 \cs_new_protected:Npn \xunadd_set_begin_hook:nn
7741 { \__xunadd_set_cmd_hook:nnn { begin } }
7742 \cs_new_protected:Npn \xunadd_set_end_hook:nn
7743 { \__xunadd_set_cmd_hook:nnn { end } }
7744 \cs_new_protected:Npn \__xunadd_set_cmd_hook:nnn #1#2#3
7745 {
7746   \cs_set_protected:cpn
7747   {
7748     \tl_if_single:nTF {#2}
7749     { \use:c { __xunadd_#1_csname:n } { \token_to_str:N #2 } }
7750     { \__xunadd_set_cmd_hook_aux:Nwn #2 \q_stop {#1} }
7751   } ##1
7752   {#3}
7753 }
7754 \cs_new:Npn \__xunadd_set_cmd_hook_aux:Nwn #1#2 \q_stop #3
7755 { \use:c { __xunadd_#3_csname:n } { \token_to_str:N #1 - \tl_to_str:n {#2} } }
7756 \cs_new:Npn \__xunadd_begin_csname:n #1 { __xunadd_begin_#1_hook:n }
7757 \cs_new:Npn \__xunadd_end_csname:n #1 { __xunadd_end_#1_hook:n }
7758 \tl_new:N \l__xunadd_begin_hook_tl
7759 \tl_new:N \l__xunadd_end_hook_tl

```

```

\__xunadd_begin_hook:nn 7760 \cs_new_protected:Npn \__xunadd_begin_hook:nn #1#2
\__xunadd_end_hook:nn 7761 {
7762   \tl_use:N \l__xunadd_begin_hook_tl
7763   \cs_if_exist_use:cF { \__xunadd_begin_csname:n { #1 - \tl_to_str:n {#2} } }
7764   { \cs_if_exist_use:cF { \__xunadd_begin_csname:n {#1} } { \use_none:n } }
7765   {#2}
7766 }
7767 \cs_new_protected:Npn \__xunadd_end_hook:nn #1#2
7768 {
7769   \cs_if_exist_use:cF { \__xunadd_end_csname:n { #1 - \tl_to_str:n {#2} } }
7770   { \cs_if_exist_use:cF { \__xunadd_end_csname:n {#1} } { \use_none:n } }
7771   {#2}
7772   \tl_use:N \l__xunadd_end_hook_tl
7773 }

\DeclareUTFTIPACCommand 7774 \NewDocumentCommand \DeclareUTFTIPACCommand { 0 { \UTFencname } m }
7775 { \use:e { \__xunadd_text_tipa_command:Nnn \exp_not:N #2 { \token_to_str:N #2 } {#1} } }
7776 \cs_new_protected:Npn \__xunadd_text_tipa_command:Nnn #1#2#3
7777 {
7778   \cs_set_eq:cc { UTF/#3#2 } { #3#2 }
7779   \DeclareTextCommand #1 {#3} { \__xunadd_text_tipa_command:nnn {#3} {#2} }
7780 }
7781 \cs_new_protected:Npn \__xunadd_text_tipa_command:nnn #1#2#3
7782 {
7783   \exp_args:Ncc \__xunadd_check_for_tipa:NNn
7784   { \use_none:n #2 } { UTF/#1#2 } {#3}
7785 }
7786 \cs_new_protected:Npn \__xunadd_check_for_tipa:NNn #1#2#3
7787 {
7788   \tl_if_head_eq_meaning:nNTF {#3} \textipa
7789   {
7790     \exp_after:wN \tipacatchonechar \exp_after:wN
7791     { \exp_after:wN #1 \use_none:n #3 }
7792   }
7793   { #2 {#3} }
7794 }

```

\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。

```

7795 \cs_new_protected:Npn \xunadd_get_slot:nn #1#2
7796 { \__xunadd_get_slot:wn #2 \q_nil \q_stop {#1} }
7797 \cs_new_protected:Npn \__xunadd_get_slot:wn #1#2#3 \q_stop #4
7798 {
7799   \int_set:Nn \l_xunadd_slot_int { -1 }
7800   \bool_set_false:N \l_xunadd_rest_bool
7801   \group_begin: \exp_args:Nccc \group_end:
7802   { \__xunadd_get_slot:NNnn }
7803   { #4 \token_to_str:N #1 }
7804   { \__xunadd_composite_cs:Nnn #1 {#4} {#2} }
7805   {#2}
7806   {#3}
7807 }
7808 \int_new:N \l_xunadd_slot_int
7809 \bool_new:N \l_xunadd_rest_bool
7810 \cs_new_protected:Npn \__xunadd_get_slot:NNnn #1#2#3#4
7811 {
7812   \cs_if_free:NF #1
7813   {
7814     \cs_if_exist:NTF #2
7815     { \__xunadd_get_composite_slot:Nn #2 {#4} }
7816     { \__xunadd_get_character_slot:Nn #1 { #3 #4 } }
7817   }
7818 }
7819 \cs_new_protected:Npn \__xunadd_get_composite_slot:Nn #1#2
7820 {
7821   \token_if_chardef:NT #1
7822   {
7823     \int_set:Nn \l_xunadd_slot_int {#1}

```

```

7824     \quark_if_nil:nF {#2}
7825     { \bool_set_true:N \l_xunadd_rest_bool }
7826   }
7827 }
7828 \cs_new_protected:Npn \__xunadd_get_character_slot:Nn #1
7829 {
7830   \exp_after:wN \__xunadd_get_character_slot_aux:wn #1
7831   \__xunadd_text_character:nN \q_nil \q_nil \q_stop
7832 }
7833 \cs_new_protected:Npn \__xunadd_get_character_slot_aux:wn
7834 #1 \__xunadd_text_character:nN #2#3#4 \q_stop #5
7835 {
7836   \quark_if_nil:nF {#2}
7837   {
7838     \int_set:Nn \l_xunadd_slot_int { `#3 }
7839     \quark_if_nil:nF {#5}
7840     { \bool_set_true:N \l_xunadd_rest_bool }
7841   }
7842 }

```

\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。

```

7843 \cs_new_protected_nopar:Npn \xunadd@microtype@is@charx #1 \relax
7844 {
7845   \use:e
7846   { \xunadd_get_slot:nn { \MT@encoding } { \tex_the:D \MT@toks } }
7847   \int_compare:nNnTF \l_xunadd_slot_int < \c_zero_int
7848   { \xunadd@original@is@charx #1 \relax }
7849   {
7850     \cs_set_nopar:Npe \MT@char@ { \int_use:N \l_xunadd_slot_int }
7851     \bool_if:NT \l_xunadd_rest_bool { \MT@norestfalse }
7852   }
7853 }
7854 \cs_new_protected_nopar:Npn \xunadd@microtype@hook
7855 {
7856   \cs_if_free:NF \MT@is@charx
7857   {
7858     \cs_new_eq:NN \xunadd@original@is@charx \MT@is@charx
7859     \cs_set_eq:NN \MT@is@charx \xunadd@microtype@is@charx
7860     \cs_set_eq:NN \MT@warn@unknown@once \use_none:n
7861   }
7862 }
7863 \@ifpackageloaded { microtype }
7864 { \use:n } { \AtBeginDocument }
7865 { \xunadd@microtype@hook }

7866 </xunicode>
7867 <*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。

```

7868 \DeclareUTFSymbol\texthyphenationpoint{"2027}
7869 \DeclareUTFSymbol\textwoemdash{"2E3A}

```

以下内容选自 xunicode, 并做了适当修改。

```

7870 \DeclareUTFComposite\textsuperscript
7871 \DeclareUTFComposite\textsubscript
7872 \DeclareUTFEncodedAccent\textsbleftarrow{"20EE}{ "02FF}
7873 \DeclareUTFEncodedAccent\`{"0300}{ "02CB}
7874 \DeclareUTFEncodedAccent\capitalgrave{"0300}{ "02CB}
7875 \DeclareUTFEncodedAccent\`{"0301}{ "02CA}
7876 \DeclareUTFEncodedAccent\capitalacute{"0301}{ "02CA}
7877 \DeclareUTFEncodedAccent\`{"0302}{ "02C6}
7878 \DeclareUTFEncodedAccent\capitalcircumflex{"0302}{ "02C6}
7879 \DeclareUTFEncodedAccent\`{"0303}{ "02DC}

```

```

7880 \DeclareUTFEncodedAccent\capitaltilde{"0303"}{"02DC}
7881 \DeclareUTFEncodedAccent\={{"0304"}{"02C9}
7882 \DeclareUTFEncodedAccent\capitalmacron{"0304"}{"02C9}
7883 \DeclareUTFEncodedAccent\textoverline{"0305"}{"203E}
7884 \DeclareUTFEncodedAccent\u{"0306"}{"02D8}
7885 \DeclareUTFEncodedAccent\capitalbreve{"0306"}{"02D8}
7886 \DeclareUTFEncodedAccent\{"0307"}{"02D9}
7887 \DeclareUTFEncodedAccent\capitaldotaccent{"0307"}{"02D9}
7888 \DeclareUTFEncodedAccent\{"0308"}{"00A8}
7889 \DeclareUTFEncodedAccent\capitaldieresis{"0308"}{"00A8}
7890 \DeclareUTFEncodedAccent\m{"0309"}{"0309}
7891 \DeclareUTFEncodedAccent\texthookabove{"0309"}{"0309}
7892 \DeclareUTFEncodedAccent\r{"030A"}{"02DA}
7893 \DeclareUTFEncodedAccent\capitalring{"030A"}{"02DA}
7894 \DeclareUTFEncodedAccent\H{"030B"}{"02DD}
7895 \DeclareUTFEncodedAccent\capitalhungarumlaut{"030B"}{"02DD}
7896 \DeclareUTFEncodedAccent\v{"030C"}{"02C7}
7897 \DeclareUTFEncodedAccent\capitalcaron{"030C"}{"02C7}
7898 \DeclareUTFEncodedAccent\textvbaraccent{"030D"}{"02C8}
7899 \DeclareUTFEncodedAccent\textdoublevbaraccent{"030E"}{"0022}
7900 \DeclareUTFEncodedAccent\U{"030E"}{"0022}
7901 \DeclareUTFEncodedAccent\textdoublegrave{"030F"}{"02F5}
7902 \DeclareUTFEncodedAccent\G{"030F"}{"02F5}
7903 \DeclareUTFEncodedAccent\textdotbreve{"0310"}{"0310}
7904 \DeclareUTFEncodedAccent\textroundcap{"0311"}{"0311}
7905 \DeclareUTFEncodedAccent\newtie{"0311"}{"0311}
7906 \DeclareUTFEncodedAccent\capitalnewtie{"0311"}{"0311}
7907 \DeclareUTFEncodedAccent\textturncommaabove{"0312"}{"02BB}
7908 \DeclareUTFEncodedAccent\textcommaabove{"0313"}{"02BC}
7909 \DeclareUTFEncodedAccent\textrevcommaabove{"0314"}{"02BD}
7910 \DeclareUTFEncodedAccent\overbridge{"0346"}{"0346}
7911 \DeclareUTFEncodedAccent\crtilde{"034A"}{"034A}
7912 \DeclareUTFEncodedAccent\dottedtilde{"034B"}{"034B}
7913 \DeclareUTFEncodedAccent\doubletilde{"034C"}{"034C}
7914 \DeclareUTFEncodedAccent\textrightarrowhead{"0350"}{"02C3}
7915 \DeclareUTFEncodedAccent\textlefthalfing{"0351"}{"02D3}
7916 \DeclareUTFEncodedAccent\texttrighthalfing{"0357"}{"02D2}
7917 \DeclareUTFDoubleEncodedSymbol\textdoublebrevebelow{"035C"}{"035C}
7918 \DeclareUTFDoubleEncodedAccent\textdoublebreve{"035D"}{"035D}
7919 \DeclareUTFDoubleEncodedAccent\textdoublemacron{"035E"}{"035E}
7920 \DeclareUTFDoubleEncodedSymbol\textdoublemacronbelow{"035F"}{"035F}
7921 \DeclareUTFDoubleEncodedAccent\textdoubletilde{"0360"}{"0360}
7922 \DeclareUTFDoubleEncodedAccent\t{"0361"}{"0361}
7923 \DeclareUTFDoubleEncodedAccent\capitaltie{"0361"}{"0361}
7924 \DeclareUTFDoubleEncodedAccent\texttoptiebar{"0361"}{"0361}
7925 \DeclareUTFDoubleEncodedSymbol\sliding{"0362"}{"0362}
7926 \DeclareUTFTIPACCommand\t
7927 \DeclareUTFTIPACCommand\capitaltie
7928 \DeclareUTFTIPACCommand\texttoptiebar
7929 \DeclareUTFTIPACCommand\sliding
7930 \DeclareUTFEncodedAccent\texthighrise{"1DC4"}{"1DC4}
7931 \DeclareUTFEncodedAccent\textlowrise{"1DC5"}{"1DC5}
7932 \DeclareUTFEncodedAccent\textrisefall{"1DC8"}{"1DC8}
7933 \DeclareUTFEncodedAccent\textfallrise{"1DC9"}{"1DC9}
7934 \DeclareUTFEncodedAccent\texttaolig{"1DD5"}{"1DD5}
7935 \DeclareUTFCompositeSymbol\textundertie{H}{"1E2A}
7936 \DeclareUTFCompositeSymbol\textundertie{h}{"1E2B}
7937 \DeclareUTFEncodedAccents\textcircumgrave{"0302"}{"0301}
7938 \DeclareUTFSymbol\textFinv{"2132}
7939 \DeclareUTFSymbol\textaleph{"2135}
7940 \DeclareUTFSymbol\textbeth{"2136}
7941 \DeclareUTFSymbol\textgimel{"2137}
7942 \DeclareUTFSymbol\textdaleth{"2138}
7943 \DeclareUTFSymbol\textGame{"2141}
7944 \DeclareUTFCompositeCommand\tonebar{25}{\tonebar{2}\tonebar{5}}
7945 \DeclareUTFCompositeCommand\tonebar{52}{\tonebar{5}\tonebar{2}}
7946 \DeclareUTFSymbol\textbigcircle{"25EF}

```

```

7947 \DeclareUTFEncodedCircle\textcircled{"20DD"}{"25EF}
7948 \DeclareUTFCompositeSymbol\textcircled{0}{"24EA}
7949 \DeclareUTFCompositeSymbol\textcircled{1}{"2460}
7950 \DeclareUTFCompositeSymbol\textcircled{2}{"2461}
7951 \DeclareUTFCompositeSymbol\textcircled{3}{"2462}
7952 \DeclareUTFCompositeSymbol\textcircled{4}{"2463}
7953 \DeclareUTFCompositeSymbol\textcircled{5}{"2464}
7954 \DeclareUTFCompositeSymbol\textcircled{6}{"2465}
7955 \DeclareUTFCompositeSymbol\textcircled{7}{"2466}
7956 \DeclareUTFCompositeSymbol\textcircled{8}{"2467}
7957 \DeclareUTFCompositeSymbol\textcircled{9}{"2468}
7958 \DeclareUTFCompositeSymbol\textcircled{10}{"2469}
7959 \DeclareUTFCompositeSymbol\textcircled{11}{"246A}
7960 \DeclareUTFCompositeSymbol\textcircled{12}{"246B}
7961 \DeclareUTFCompositeSymbol\textcircled{13}{"246C}
7962 \DeclareUTFCompositeSymbol\textcircled{14}{"246D}
7963 \DeclareUTFCompositeSymbol\textcircled{15}{"246E}
7964 \DeclareUTFCompositeSymbol\textcircled{16}{"246F}
7965 \DeclareUTFCompositeSymbol\textcircled{17}{"2470}
7966 \DeclareUTFCompositeSymbol\textcircled{18}{"2471}
7967 \DeclareUTFCompositeSymbol\textcircled{19}{"2472}
7968 \DeclareUTFCompositeSymbol\textcircled{20}{"2473}
7969 \DeclareUTFCompositeSymbol\textcircled{21}{"3251}
7970 \DeclareUTFCompositeSymbol\textcircled{22}{"3252}
7971 \DeclareUTFCompositeSymbol\textcircled{23}{"3253}
7972 \DeclareUTFCompositeSymbol\textcircled{24}{"3254}
7973 \DeclareUTFCompositeSymbol\textcircled{25}{"3255}
7974 \DeclareUTFCompositeSymbol\textcircled{26}{"3256}
7975 \DeclareUTFCompositeSymbol\textcircled{27}{"3257}
7976 \DeclareUTFCompositeSymbol\textcircled{28}{"3258}
7977 \DeclareUTFCompositeSymbol\textcircled{29}{"3259}
7978 \DeclareUTFCompositeSymbol\textcircled{30}{"325A}
7979 \DeclareUTFCompositeSymbol\textcircled{31}{"325B}
7980 \DeclareUTFCompositeSymbol\textcircled{32}{"325C}
7981 \DeclareUTFCompositeSymbol\textcircled{33}{"325D}
7982 \DeclareUTFCompositeSymbol\textcircled{34}{"325E}
7983 \DeclareUTFCompositeSymbol\textcircled{35}{"325F}
7984 \DeclareUTFCompositeSymbol\textcircled{36}{"32B1}
7985 \DeclareUTFCompositeSymbol\textcircled{37}{"32B2}
7986 \DeclareUTFCompositeSymbol\textcircled{38}{"32B3}
7987 \DeclareUTFCompositeSymbol\textcircled{39}{"32B4}
7988 \DeclareUTFCompositeSymbol\textcircled{40}{"32B5}
7989 \DeclareUTFCompositeSymbol\textcircled{41}{"32B6}
7990 \DeclareUTFCompositeSymbol\textcircled{42}{"32B7}
7991 \DeclareUTFCompositeSymbol\textcircled{43}{"32B8}
7992 \DeclareUTFCompositeSymbol\textcircled{44}{"32B9}
7993 \DeclareUTFCompositeSymbol\textcircled{45}{"32BA}
7994 \DeclareUTFCompositeSymbol\textcircled{46}{"32BB}
7995 \DeclareUTFCompositeSymbol\textcircled{47}{"32BC}
7996 \DeclareUTFCompositeSymbol\textcircled{48}{"32BD}
7997 \DeclareUTFCompositeSymbol\textcircled{49}{"32BE}
7998 \DeclareUTFCompositeSymbol\textcircled{50}{"32BF}
7999 \DeclareUTFCompositeSymbol\textcircled{A}{"24B6}
8000 \DeclareUTFCompositeSymbol\textcircled{B}{"24B7}
8001 \DeclareUTFCompositeSymbol\textcircled{C}{"24B8}
8002 \DeclareUTFCompositeSymbol\textcircled{D}{"24B9}
8003 \DeclareUTFCompositeSymbol\textcircled{E}{"24BA}
8004 \DeclareUTFCompositeSymbol\textcircled{F}{"24BB}
8005 \DeclareUTFCompositeSymbol\textcircled{G}{"24BC}
8006 \DeclareUTFCompositeSymbol\textcircled{H}{"24BD}
8007 \DeclareUTFCompositeSymbol\textcircled{I}{"24BE}
8008 \DeclareUTFCompositeSymbol\textcircled{J}{"24BF}
8009 \DeclareUTFCompositeSymbol\textcircled{K}{"24C0}
8010 \DeclareUTFCompositeSymbol\textcircled{L}{"24C1}
8011 \DeclareUTFCompositeSymbol\textcircled{M}{"24C2}
8012 \DeclareUTFCompositeSymbol\textcircled{N}{"24C3}
8013 \DeclareUTFCompositeSymbol\textcircled{O}{"24C4}

```

```

8014 \DeclareUTFCompositeSymbol\textcircled{P}{\textcircled{P}}{"24C5}
8015 \DeclareUTFCompositeSymbol\textcircled{Q}{\textcircled{Q}}{"24C6}
8016 \DeclareUTFCompositeSymbol\textcircled{R}{\textcircled{R}}{"24C7}
8017 \DeclareUTFCompositeSymbol\textcircled{S}{\textcircled{S}}{"24C8}
8018 \DeclareUTFCompositeSymbol\textcircled{T}{\textcircled{T}}{"24C9}
8019 \DeclareUTFCompositeSymbol\textcircled{U}{\textcircled{U}}{"24CA}
8020 \DeclareUTFCompositeSymbol\textcircled{V}{\textcircled{V}}{"24CB}
8021 \DeclareUTFCompositeSymbol\textcircled{W}{\textcircled{W}}{"24CC}
8022 \DeclareUTFCompositeSymbol\textcircled{X}{\textcircled{X}}{"24CD}
8023 \DeclareUTFCompositeSymbol\textcircled{Y}{\textcircled{Y}}{"24CE}
8024 \DeclareUTFCompositeSymbol\textcircled{Z}{\textcircled{Z}}{"24CF}
8025 \DeclareUTFCompositeSymbol\textcircled{a}{\textcircled{a}}{"24D0}
8026 \DeclareUTFCompositeSymbol\textcircled{b}{\textcircled{b}}{"24D1}
8027 \DeclareUTFCompositeSymbol\textcircled{c}{\textcircled{c}}{"24D2}
8028 \DeclareUTFCompositeSymbol\textcircled{d}{\textcircled{d}}{"24D3}
8029 \DeclareUTFCompositeSymbol\textcircled{e}{\textcircled{e}}{"24D4}
8030 \DeclareUTFCompositeSymbol\textcircled{f}{\textcircled{f}}{"24D5}
8031 \DeclareUTFCompositeSymbol\textcircled{g}{\textcircled{g}}{"24D6}
8032 \DeclareUTFCompositeSymbol\textcircled{h}{\textcircled{h}}{"24D7}
8033 \DeclareUTFCompositeSymbol\textcircled{i}{\textcircled{i}}{"24D8}
8034 \DeclareUTFCompositeSymbol\textcircled{j}{\textcircled{j}}{"24D9}
8035 \DeclareUTFCompositeSymbol\textcircled{k}{\textcircled{k}}{"24DA}
8036 \DeclareUTFCompositeSymbol\textcircled{l}{\textcircled{l}}{"24DB}
8037 \DeclareUTFCompositeSymbol\textcircled{m}{\textcircled{m}}{"24DC}
8038 \DeclareUTFCompositeSymbol\textcircled{n}{\textcircled{n}}{"24DD}
8039 \DeclareUTFCompositeSymbol\textcircled{o}{\textcircled{o}}{"24DE}
8040 \DeclareUTFCompositeSymbol\textcircled{p}{\textcircled{p}}{"24DF}
8041 \DeclareUTFCompositeSymbol\textcircled{q}{\textcircled{q}}{"24E0}
8042 \DeclareUTFCompositeSymbol\textcircled{r}{\textcircled{r}}{"24E1}
8043 \DeclareUTFCompositeSymbol\textcircled{s}{\textcircled{s}}{"24E2}
8044 \DeclareUTFCompositeSymbol\textcircled{t}{\textcircled{t}}{"24E3}
8045 \DeclareUTFCompositeSymbol\textcircled{u}{\textcircled{u}}{"24E4}
8046 \DeclareUTFCompositeSymbol\textcircled{v}{\textcircled{v}}{"24E5}
8047 \DeclareUTFCompositeSymbol\textcircled{w}{\textcircled{w}}{"24E6}
8048 \DeclareUTFCompositeSymbol\textcircled{x}{\textcircled{x}}{"24E7}
8049 \DeclareUTFCompositeSymbol\textcircled{y}{\textcircled{y}}{"24E8}
8050 \DeclareUTFCompositeSymbol\textcircled{z}{\textcircled{z}}{"24E9}
8051 \DeclareUTFCompositeSymbol\textsuperscript{h}{\textsuperscript{h}}{"02B0}
8052 \DeclareUTFCompositeSymbol\textsuperscript{\textth}{\textth}{"02B1}
8053 \DeclareUTFCompositeSymbol\textsuperscript{j}{\textsuperscript{j}}{"02B2}
8054 \DeclareUTFCompositeSymbol\textsuperscript{r}{\textsuperscript{r}}{"02B3}
8055 \DeclareUTFCompositeSymbol\textsuperscript{\textturnr}{\textturnr}{"02B4}
8056 \DeclareUTFCompositeSymbol\textsuperscript{\textturnrrtail}{\textturnrrtail}{"02B5}
8057 \DeclareUTFCompositeSymbol\textsuperscript{\textinvscr}{\textinvscr}{"02B6}
8058 \DeclareUTFCompositeSymbol\textsuperscript{w}{\textsuperscript{w}}{"02B7}
8059 \DeclareUTFCompositeSymbol\textsuperscript{y}{\textsuperscript{y}}{"02B8}
8060 \DeclareUTFCompositeSymbol\textsuperscript{\textbabygamma}{\textbabygamma}{"02E0}
8061 \DeclareUTFCompositeSymbol\textsuperscript{\textgammalatinsmall}{\textgammalatinsmall}{"02E0}
8062 \DeclareUTFCompositeSymbol\textsuperscript{l}{\textsuperscript{l}}{"02E1}
8063 \DeclareUTFCompositeSymbol\textsuperscript{s}{\textsuperscript{s}}{"02E2}
8064 \DeclareUTFCompositeSymbol\textsuperscript{x}{\textsuperscript{x}}{"02E3}
8065 \DeclareUTFCompositeSymbol\textsuperscript{\textrevglotstop}{\textrevglotstop}{"02E4}
8066 \DeclareUTFCompositeSymbol\textsuperscript{\textrevepsilon}{\textrevepsilon}{"1D4C}
8067 \DeclareUTFCompositeSymbol\textsuperscript{cyrn}{\textsuperscript{cyrn}}{"1D78}
8068 \DeclareUTFCompositeSymbol\textsuperscript{\textbarsci}{\textbarsci}{"1DA7}
8069 \DeclareUTFCompositeSymbol\textsuperscript{V}{\textsuperscript{V}}{"2C7D}
8070 \DeclareUTFCompositeSymbol\textsuperscript{\textHbar}{\textHbar}{"A7F8}
8071 \DeclareUTFCompositeSymbol\textsuperscript{\textHslash}{\textHslash}{"A7F8}
8072 \DeclareUTFCompositeSymbol\textsuperscript{oe}{\textsuperscript{oe}}{"A7F9}
8073 \DeclareUTFCompositeSymbol\textsubscript{h}{\textsubscript{h}}{"2095}
8074 \DeclareUTFCompositeSymbol\textsubscript{k}{\textsubscript{k}}{"2096}
8075 \DeclareUTFCompositeSymbol\textsubscript{l}{\textsubscript{l}}{"2097}
8076 \DeclareUTFCompositeSymbol\textsubscript{m}{\textsubscript{m}}{"2098}
8077 \DeclareUTFCompositeSymbol\textsubscript{n}{\textsubscript{n}}{"2099}
8078 \DeclareUTFCompositeSymbol\textsubscript{p}{\textsubscript{p}}{"209A}
8079 \DeclareUTFCompositeSymbol\textsubscript{s}{\textsubscript{s}}{"209B}
8080 \DeclareUTFCompositeSymbol\textsubscript{t}{\textsubscript{t}}{"209C}

```

以下定义取自 `hyperref` 的 `puenc.def`。

```

8081 \DeclareUTFEncodedAccent\textinvbreve{"0311}{\textinverted{0311}}
8082 \DeclareUTFEncodedSymbol\textsubbreve{"032E}{\textsubscript{032E}}
8083 \DeclareUTFSymbol\textHT{"0009}
8084 \DeclareUTFSymbol\textLF{"000A}
8085 \DeclareUTFSymbol\textCR{"000D}
8086 \DeclareUTFSymbol\textnumbersign{"0023}
8087 \DeclareUTFSymbol\textparenleft{"0028}
8088 \DeclareUTFSymbol\textparenright{"0029}
8089 \DeclareUTFSymbol\textMVPlus{"002B}
8090 \DeclareUTFSymbol\textMVComma{"002C}
8091 \DeclareUTFSymbol\textMVMinus{"002D}
8092 \DeclareUTFSymbol\textMVPeriod{"002E}
8093 \DeclareUTFSymbol\textMVDivision{"002F}
8094 \DeclareUTFSymbol\textMVZero{"0030}
8095 \DeclareUTFSymbol\textMVOne{"0031}
8096 \DeclareUTFSymbol\textMVTwo{"0032}
8097 \DeclareUTFSymbol\textMVThree{"0033}
8098 \DeclareUTFSymbol\textMVFour{"0034}
8099 \DeclareUTFSymbol\textMVFive{"0035}
8100 \DeclareUTFSymbol\textMVSix{"0036}
8101 \DeclareUTFSymbol\textMVSeven{"0037}
8102 \DeclareUTFSymbol\textMVEight{"0038}
8103 \DeclareUTFSymbol\textMVNine{"0039}
8104 \DeclareUTFSymbol\textMVAt{"0040}
8105 \DeclareUTFCompositeCommand\.\{i}{i}
8106 \DeclareUTFCompositeCommand\.\{i}{i}
8107 \DeclareUTFSymbol\textlnot{"00AC}
8108 \DeclareUTFSymbol\textplusminus{"00B1}
8109 \DeclareUTFSymbol\textcedilla{"00B8}
8110 \DeclareUTFSymbol\textmultiply{"00D7}
8111 \DeclareUTFSymbol\textThorn{"00DE}
8112 \DeclareUTFSymbol\textdivide{"00F7}
8113 \DeclareUTFSymbol\textHslash{"0126}
8114 \DeclareUTFCompositeSymbol\k{i}{i}{012F}
8115 \DeclareUTFCompositeSymbol\.\{L}{L}{013F}
8116 \DeclareUTFCompositeSymbol\.\{l}{l}{0140}
8117 \DeclareUTFSymbol\textnapostrophe{"0149}
8118 \DeclareUTFSymbol\textTslash{"0166}
8119 \DeclareUTFSymbol\texttslash{"0167}
8120 \DeclareUTFSymbol\textlongS{"017F}
8121 \DeclareUTFSymbol\textthausaB{"0181}
8122 \DeclareUTFSymbol\textthausaD{"018A}
8123 \DeclareUTFSymbol\textrevE{"018E}
8124 \DeclareUTFSymbol\textthausaK{"0198}
8125 \DeclareUTFSymbol\textPUnrleg{"019E}
8126 \DeclareUTFSymbol\textinve{"01DD}
8127 \DeclareUTFSymbol\textGslash{"01E4}
8128 \DeclareUTFSymbol\textgslash{"01E5}
8129 \DeclareUTFCompositeSymbol\textinvbreve{E}{E}{0206}
8130 \DeclareUTFCompositeSymbol\textinvbreve{e}{e}{0207}
8131 \DeclareUTFCompositeSymbol\textinvbreve{I}{I}{020A}
8132 \DeclareUTFCompositeSymbol\textinvbreve{i}{i}{020B}
8133 \DeclareUTFCompositeSymbol\textinvbreve{\i}{\i}{020B}
8134 \DeclareUTFCompositeSymbol\textinvbreve{O}{O}{020E}
8135 \DeclareUTFCompositeSymbol\textinvbreve{o}{o}{020F}
8136 \DeclareUTFCompositeSymbol\textinvbreve{U}{U}{0216}
8137 \DeclareUTFCompositeSymbol\textinvbreve{u}{u}{0217}
8138 \DeclareUTFSymbol\j{"0237}
8139 \DeclareUTFSymbol\textPUdblig{"0238}
8140 \DeclareUTFSymbol\textPUqplig{"0239}
8141 \DeclareUTFSymbol\textslashc{"023C}
8142 \DeclareUTFSymbol\textniepsilon{"025B}
8143 \DeclareUTFSymbol\textipagamma{"0263}
8144 \DeclareUTFSymbol\textniiota{"0269}
8145 \DeclareUTFSymbol\textniiota{"0269}
8146 \DeclareUTFSymbol\textniupsilon{"028A}

```

```

8147 \DeclareUTFSymbol\textstring{"02DA}
8148 \DeclareUTFSymbol\texttilde{"02DC}
8149 \DeclareUTFSymbol\texthungarumlaut{"02DD}
8150 \DeclareUTFSymbol\textstringlow{"02F3}
8151 \DeclareUTFSymbol\texttildebelow{"02F7}
8152 \DeclareUTFCommand\textnewtie{\textinvbreve\ }
8153 \DeclareUTFCommand\textdotbelow{\d\ }
8154 \DeclareUTFSymbol\textmacronbelow{"02CD}
8155 \DeclareUTFCommand\texttie{\t\ }
8156 \DeclareUTFSymbol\textnumeralsigngreek{"0374}
8157 \DeclareUTFSymbol\textnumeralsignlowergreek{"0375}
8158 \DeclareUTFCompositeSymbol\{'\textAlpha>{"0386}
8159 \DeclareUTFCompositeSymbol\{'\textEpsilon>{"0388}
8160 \DeclareUTFCompositeSymbol\{'\textEta>{"0389}
8161 \DeclareUTFCompositeSymbol\{'\textIota>{"038A}
8162 \DeclareUTFCompositeSymbol\{'\textOmicron>{"038C}
8163 \DeclareUTFCompositeSymbol\{'\textUpsilon>{"038E}
8164 \DeclareUTFCompositeSymbol\{'\textOmega>{"038F}
8165 \DeclareUTFCompositeSymbol\{'\textIotadieresis>{"0390}
8166 \DeclareUTFSymbol\textIotadieresis{"03AA}
8167 \DeclareUTFCompositeSymbol\{"\textIota>{"03AA}
8168 \DeclareUTFCompositeSymbol\{"\textUpsilon>{"03AB}
8169 \DeclareUTFCompositeSymbol\{'\textalpha>{"03AC}
8170 \DeclareUTFCompositeSymbol\{'\textepsilon>{"03AD}
8171 \DeclareUTFCompositeSymbol\{'\texteta>{"03AE}
8172 \DeclareUTFCompositeSymbol\{'\textiota>{"03AF}
8173 \DeclareUTFCompositeSymbol\{"\textupsilonacute>{"03B0}
8174 \DeclareUTFSymbol\textmugreek{"03BC}
8175 \DeclareUTFSymbol\textvarsigma{"03C2}
8176 \DeclareUTFCompositeSymbol\{"\textiota>{"03CA}
8177 \DeclareUTFCompositeSymbol\{"\textupsilon>{"03CB}
8178 \DeclareUTFCompositeSymbol\{'\textomicron>{"03CC}
8179 \DeclareUTFSymbol\textupsilonacute{"03CD}
8180 \DeclareUTFCompositeSymbol\{'\textupsilon>{"03CD}
8181 \DeclareUTFCompositeSymbol\{'\textomega>{"03CE}
8182 \DeclareUTFSymbol\textStigmatreek{"03DA}
8183 \DeclareUTFSymbol\textstigmatreek{"03DB}
8184 \DeclareUTFSymbol\textDigammagreek{"03DC}
8185 \DeclareUTFSymbol\textdigammagreek{"03DD}
8186 \DeclareUTFSymbol\textKoppagreek{"03DE}
8187 \DeclareUTFSymbol\textkoppagreek{"03DF}
8188 \DeclareUTFSymbol\textSampigreek{"03E0}
8189 \DeclareUTFSymbol\textsampigreek{"03E1}
8190 \DeclareUTFSymbol\textbackepsilon{"03F6}
8191 \DeclareUTFCompositeSymbol\{'\CYRE>{"0400}
8192 \DeclareUTFSymbol\CYRYO{"0401}
8193 \DeclareUTFCompositeSymbol\{"\CYRE>{"0401}
8194 \DeclareUTFSymbol\CYRDJE{"0402}
8195 \DeclareUTFCompositeSymbol\{'\CYRG>{"0403}
8196 \DeclareUTFSymbol\CYRIE{"0404}
8197 \DeclareUTFSymbol\CYRDZE{"0405}
8198 \DeclareUTFSymbol\CYRII{"0406}
8199 \DeclareUTFSymbol\CYRYI{"0407}
8200 \DeclareUTFCompositeSymbol\{"\CYRII>{"0407}
8201 \DeclareUTFSymbol\CYRJE{"0408}
8202 \DeclareUTFSymbol\CYRLJE{"0409}
8203 \DeclareUTFSymbol\CYRNJE{"040A}
8204 \DeclareUTFSymbol\CYRTSHE{"040B}
8205 \DeclareUTFCompositeSymbol\{'\CYRK>{"040C}
8206 \DeclareUTFCompositeSymbol\{'\CYRI>{"040D}
8207 \DeclareUTFSymbol\CYRUSHRT{"040E}
8208 \DeclareUTFCompositeSymbol\U{\CYRU>{"040E}
8209 \DeclareUTFSymbol\CYRDZHE{"040F}
8210 \DeclareUTFSymbol\CYRA{"0410}
8211 \DeclareUTFSymbol\CYRB{"0411}
8212 \DeclareUTFSymbol\CYRV{"0412}
8213 \DeclareUTFSymbol\CYRG{"0413}

```

```

8214 \DeclareUTFSymbol\CYRD{"0414}
8215 \DeclareUTFSymbol\CYRE{"0415}
8216 \DeclareUTFSymbol\CYRZH{"0416}
8217 \DeclareUTFSymbol\CYRZ{"0417}
8218 \DeclareUTFSymbol\CYRI{"0418}
8219 \DeclareUTFSymbol\CYRISHRT{"0419}
8220 \DeclareUTFCompositeSymbol\U{\CYRI}{{"0419}
8221 \DeclareUTFSymbol\CYRK{"041A}
8222 \DeclareUTFSymbol\CYRL{"041B}
8223 \DeclareUTFSymbol\CYRM{"041C}
8224 \DeclareUTFSymbol\CYRN{"041D}
8225 \DeclareUTFSymbol\CYRO{"041E}
8226 \DeclareUTFSymbol\CYRP{"041F}
8227 \DeclareUTFSymbol\CYRR{"0420}
8228 \DeclareUTFSymbol\CYRS{"0421}
8229 \DeclareUTFSymbol\CYRT{"0422}
8230 \DeclareUTFSymbol\CYRU{"0423}
8231 \DeclareUTFSymbol\CYRF{"0424}
8232 \DeclareUTFSymbol\CYRH{"0425}
8233 \DeclareUTFSymbol\CYRC{"0426}
8234 \DeclareUTFSymbol\CYRCH{"0427}
8235 \DeclareUTFSymbol\CYRSH{"0428}
8236 \DeclareUTFSymbol\CYRSHCH{"0429}
8237 \DeclareUTFSymbol\CYRHRDSN{"042A}
8238 \DeclareUTFSymbol\CYRERY{"042B}
8239 \DeclareUTFSymbol\CYRSFTSN{"042C}
8240 \DeclareUTFSymbol\CYREREV{"042D}
8241 \DeclareUTFSymbol\CYRYU{"042E}
8242 \DeclareUTFSymbol\CYRYA{"042F}
8243 \DeclareUTFSymbol\cyra{"0430}
8244 \DeclareUTFSymbol\cyrb{"0431}
8245 \DeclareUTFSymbol\cyrv{"0432}
8246 \DeclareUTFSymbol\cyrg{"0433}
8247 \DeclareUTFSymbol\cyrd{"0434}
8248 \DeclareUTFSymbol\cyre{"0435}
8249 \DeclareUTFSymbol\cyrzh{"0436}
8250 \DeclareUTFSymbol\cyrz{"0437}
8251 \DeclareUTFSymbol\cyri{"0438}
8252 \DeclareUTFSymbol\cyrishrt{"0439}
8253 \DeclareUTFCompositeSymbol\U{\cyri}{{"0439}
8254 \DeclareUTFSymbol\cyrk{"043A}
8255 \DeclareUTFSymbol\cyr1{"043B}
8256 \DeclareUTFSymbol\cyrm{"043C}
8257 \DeclareUTFSymbol\cyrn{"043D}
8258 \DeclareUTFSymbol\cyro{"043E}
8259 \DeclareUTFSymbol\cyrp{"043F}
8260 \DeclareUTFSymbol\cyrr{"0440}
8261 \DeclareUTFSymbol\cyrs{"0441}
8262 \DeclareUTFSymbol\cyrt{"0442}
8263 \DeclareUTFSymbol\cyru{"0443}
8264 \DeclareUTFSymbol\cyrf{"0444}
8265 \DeclareUTFSymbol\cyrh{"0445}
8266 \DeclareUTFSymbol\cyrc{"0446}
8267 \DeclareUTFSymbol\cyrch{"0447}
8268 \DeclareUTFSymbol\cyrsh{"0448}
8269 \DeclareUTFSymbol\cyrshch{"0449}
8270 \DeclareUTFSymbol\cyrhrdsn{"044A}
8271 \DeclareUTFSymbol\cyrery{"044B}
8272 \DeclareUTFSymbol\cyrsftsn{"044C}
8273 \DeclareUTFSymbol\cyrerev{"044D}
8274 \DeclareUTFSymbol\cyryu{"044E}
8275 \DeclareUTFSymbol\cyrya{"044F}
8276 \DeclareUTFCompositeSymbol\`{\cyre}{{"0450}
8277 \DeclareUTFSymbol\cyryo{"0451}
8278 \DeclareUTFCompositeSymbol\{"{\cyre}{{"0451}
8279 \DeclareUTFSymbol\cyrdje{"0452}
8280 \DeclareUTFCompositeSymbol\'\{\cyrg}{{"0453}

```

```
8281 \DeclareUTFSymbol\cyrie{"0454}
8282 \DeclareUTFSymbol\cyrdze{"0455}
8283 \DeclareUTFSymbol\cyrii{"0456}
8284 \DeclareUTFSymbol\cyryi{"0457}
8285 \DeclareUTFCompositeSymbol\{"\cyrii}{0457}
8286 \DeclareUTFSymbol\cyrje{"0458}
8287 \DeclareUTFSymbol\cyrlje{"0459}
8288 \DeclareUTFSymbol\cyrnje{"045A}
8289 \DeclareUTFSymbol\cyrtshe{"045B}
8290 \DeclareUTFCompositeSymbol\'\{cyrk}{045C}
8291 \DeclareUTFCompositeSymbol\`\{cyri}{045D}
8292 \DeclareUTFSymbol\cyrushrt{"045E}
8293 \DeclareUTFCompositeSymbol\U{\curu}{045E}
8294 \DeclareUTFSymbol\cyrdzhe{"045F}
8295 \DeclareUTFSymbol\CYROMEGA{"0460}
8296 \DeclareUTFSymbol\cyromega{"0461}
8297 \DeclareUTFSymbol\CYRYAT{"0462}
8298 \DeclareUTFSymbol\cyr yat{"0463}
8299 \DeclareUTFSymbol\CYRIOTE{"0464}
8300 \DeclareUTFSymbol\cyriote{"0465}
8301 \DeclareUTFSymbol\CYRLYUS{"0466}
8302 \DeclareUTFSymbol\cyrlyus{"0467}
8303 \DeclareUTFSymbol\CYRIOTLYUS{"0468}
8304 \DeclareUTFSymbol\cyriotlyus{"0469}
8305 \DeclareUTFSymbol\CYRBYUS{"046A}
8306 \DeclareUTFSymbol\cyrbyus{"046B}
8307 \DeclareUTFSymbol\CYRIOTBYUS{"046C}
8308 \DeclareUTFSymbol\cyriotbyus{"046D}
8309 \DeclareUTFSymbol\CYRKSI{"046E}
8310 \DeclareUTFSymbol\cyrksi{"046F}
8311 \DeclareUTFSymbol\CYRPSI{"0470}
8312 \DeclareUTFSymbol\cyrpsi{"0471}
8313 \DeclareUTFSymbol\CYRFITA{"0472}
8314 \DeclareUTFSymbol\cyrfita{"0473}
8315 \DeclareUTFSymbol\CYRIZH{"0474}
8316 \DeclareUTFSymbol\cyrizh{"0475}
8317 \DeclareUTFCompositeSymbol\C{\CYRIZH}{0476}
8318 \DeclareUTFCompositeSymbol\C{\cyrizh}{0477}
8319 \DeclareUTFSymbol\CYRUK{"0478}
8320 \DeclareUTFSymbol\cyruk{"0479}
8321 \DeclareUTFSymbol\CYROMEGARND{"047A}
8322 \DeclareUTFSymbol\cyromegarnd{"047B}
8323 \DeclareUTFSymbol\CYROMEGATITLO{"047C}
8324 \DeclareUTFSymbol\cyromegatitlo{"047D}
8325 \DeclareUTFSymbol\CYROT{"047E}
8326 \DeclareUTFSymbol\cyrot{"047F}
8327 \DeclareUTFSymbol\CYRKOPPA{"0480}
8328 \DeclareUTFSymbol\cyrkoppa{"0481}
8329 \DeclareUTFSymbol\cyrthousands{"0482}
8330 \DeclareUTFSymbol\CYRISHRTDSC{"048A}
8331 \DeclareUTFSymbol\cyrishrtdsc{"048B}
8332 \DeclareUTFSymbol\CYRSEMISFTSN{"048C}
8333 \DeclareUTFSymbol\cyrsemisftsn{"048D}
8334 \DeclareUTFSymbol\CYRRTICK{"048E}
8335 \DeclareUTFSymbol\cyr rtick{"048F}
8336 \DeclareUTFSymbol\CYRGUP{"0490}
8337 \DeclareUTFSymbol\cyrgup{"0491}
8338 \DeclareUTFSymbol\CYRGHCRS{"0492}
8339 \DeclareUTFSymbol\cyrg hcrs{"0493}
8340 \DeclareUTFSymbol\CYRGHK{"0494}
8341 \DeclareUTFSymbol\cyrg hk{"0495}
8342 \DeclareUTFSymbol\CYRZHDSC{"0496}
8343 \DeclareUTFSymbol\cyrzh dsc{"0497}
8344 \DeclareUTFSymbol\CYRZDSC{"0498}
8345 \DeclareUTFCompositeSymbol\c{\CYRZ}{0498}
8346 \DeclareUTFSymbol\cyrz dsc{"0499}
8347 \DeclareUTFCompositeSymbol\c{\cyrz}{0499}
```

```

8348 \DeclareUTFSymbol\CYRKDSC{"049A}
8349 \DeclareUTFSymbol\cyrkds{"049B}
8350 \DeclareUTFSymbol\CYRKVCRS{"049C}
8351 \DeclareUTFSymbol\cyrkvc{"049D}
8352 \DeclareUTFSymbol\CYRKHCRC{"049E}
8353 \DeclareUTFSymbol\cyrkhc{"049F}
8354 \DeclareUTFSymbol\CYRKBEAK{"04A0}
8355 \DeclareUTFSymbol\cyrkbeak{"04A1}
8356 \DeclareUTFSymbol\CYRNDSC{"04A2}
8357 \DeclareUTFSymbol\cyrndsc{"04A3}
8358 \DeclareUTFSymbol\CYRNG{"04A4}
8359 \DeclareUTFSymbol\cyrng{"04A5}
8360 \DeclareUTFSymbol\CYRPHK{"04A6}
8361 \DeclareUTFSymbol\cyrphk{"04A7}
8362 \DeclareUTFSymbol\CYRABHHA{"04A8}
8363 \DeclareUTFSymbol\cyrabhha{"04A9}
8364 \DeclareUTFSymbol\CYRSDSC{"04AA}
8365 \DeclareUTFCompositeSymbol\k{\CYRS}{{"04AA}
8366 \DeclareUTFSymbol\cyrsdsc{"04AB}
8367 \DeclareUTFCompositeSymbol\k{\cyrs}{{"04AB}
8368 \DeclareUTFSymbol\CYRTDSC{"04AC}
8369 \DeclareUTFSymbol\cyrtddsc{"04AD}
8370 \DeclareUTFSymbol\CYRY{"04AE}
8371 \DeclareUTFSymbol\cyry{"04AF}
8372 \DeclareUTFSymbol\CYRYHCRC{"04B0}
8373 \DeclareUTFSymbol\cyryhc{"04B1}
8374 \DeclareUTFSymbol\CYRHDSC{"04B2}
8375 \DeclareUTFSymbol\cyrhds{"04B3}
8376 \DeclareUTFSymbol\CYRTETSE{"04B4}
8377 \DeclareUTFSymbol\cyrtetse{"04B5}
8378 \DeclareUTFSymbol\CYRCHRDSC{"04B6}
8379 \DeclareUTFSymbol\cyrchrdsc{"04B7}
8380 \DeclareUTFSymbol\CYRCHVCRS{"04B8}
8381 \DeclareUTFSymbol\cyrchvc{"04B9}
8382 \DeclareUTFSymbol\CYRSHHA{"04BA}
8383 \DeclareUTFSymbol\cyrshha{"04BB}
8384 \DeclareUTFSymbol\CYRABHCH{"04BC}
8385 \DeclareUTFSymbol\cyrabhch{"04BD}
8386 \DeclareUTFSymbol\CYRABHCHDSC{"04BE}
8387 \DeclareUTFCompositeSymbol\k{\CYRABHCH}{{"04BE}
8388 \DeclareUTFSymbol\cyrabhchdsc{"04BF}
8389 \DeclareUTFCompositeSymbol\k{\cyrabhch}{{"04BF}
8390 \DeclareUTFSymbol\CYRpalochka{"04C0}
8391 \DeclareUTFCompositeSymbol\U{\CYRZH}{{"04C1}
8392 \DeclareUTFCompositeSymbol\U{\cyrzh}{{"04C2}
8393 \DeclareUTFSymbol\CYRKHK{"04C3}
8394 \DeclareUTFSymbol\cyrkhk{"04C4}
8395 \DeclareUTFSymbol\CYRLDSC{"04C5}
8396 \DeclareUTFSymbol\cyrlsdc{"04C6}
8397 \DeclareUTFSymbol\CYRNHK{"04C7}
8398 \DeclareUTFSymbol\cyrnhk{"04C8}
8399 \DeclareUTFSymbol\CYRCHLDSC{"04CB}
8400 \DeclareUTFSymbol\cyrchldsc{"04CC}
8401 \DeclareUTFSymbol\CYRMDSC{"04CD}
8402 \DeclareUTFSymbol\cyrmdsc{"04CE}
8403 \DeclareUTFCompositeSymbol\U{\CYRA}{{"04D0}
8404 \DeclareUTFCompositeSymbol\U{\cyra}{{"04D1}
8405 \DeclareUTFCompositeSymbol\{"{\CYRA}{{"04D2}
8406 \DeclareUTFCompositeSymbol\{"{\cyra}{{"04D3}
8407 \DeclareUTFSymbol\CYRAE{"04D4}
8408 \DeclareUTFSymbol\cyrae{"04D5}
8409 \DeclareUTFCompositeSymbol\U{\CYRE}{{"04D6}
8410 \DeclareUTFCompositeSymbol\U{\cyre}{{"04D7}
8411 \DeclareUTFSymbol\CYRSCHWA{"04D8}
8412 \DeclareUTFSymbol\cyrschwa{"04D9}
8413 \DeclareUTFCompositeSymbol\{"{\CYRSCHWA}{{"04DA}
8414 \DeclareUTFCompositeSymbol\{"{\cyrschwa}{{"04DB}

```

```

8415 \DeclareUTFCompositeSymbol\{"CYRZH"}{"04DC}
8416 \DeclareUTFCompositeSymbol\{"cyrzh"}{"04DD}
8417 \DeclareUTFCompositeSymbol\{"CYRZ"}{"04DE}
8418 \DeclareUTFCompositeSymbol\{"cyrz"}{"04DF}
8419 \DeclareUTFSymbol\CYRABHDZE{"04E0}
8420 \DeclareUTFSymbol\cyrabhdze{"04E1}
8421 \DeclareUTFCompositeSymbol\={"CYRI"}{"04E2}
8422 \DeclareUTFCompositeSymbol\={"cyri"}{"04E3}
8423 \DeclareUTFCompositeSymbol\{"CYRI"}{"04E4}
8424 \DeclareUTFCompositeSymbol\{"cyri"}{"04E5}
8425 \DeclareUTFCompositeSymbol\{"CYRO"}{"04E6}
8426 \DeclareUTFCompositeSymbol\{"cyro"}{"04E7}
8427 \DeclareUTFSymbol\CYROTLD{"04E8}
8428 \DeclareUTFSymbol\cyrotld{"04E9}
8429 \DeclareUTFCompositeSymbol\{"CYROTLD"}{"04EA}
8430 \DeclareUTFCompositeSymbol\{"cyrotld"}{"04EB}
8431 \DeclareUTFCompositeSymbol\{"CYREREV"}{"04EC}
8432 \DeclareUTFCompositeSymbol\{"cyrerev"}{"04ED}
8433 \DeclareUTFCompositeSymbol\={"CYRU"}{"04EE}
8434 \DeclareUTFCompositeSymbol\={"cyru"}{"04EF}
8435 \DeclareUTFCompositeSymbol\{"CYRU"}{"04F0}
8436 \DeclareUTFCompositeSymbol\{"cyru"}{"04F1}
8437 \DeclareUTFCompositeSymbol\H{"CYRU"}{"04F2}
8438 \DeclareUTFCompositeSymbol\H{"cyru"}{"04F3}
8439 \DeclareUTFCompositeSymbol\{"CYRCH"}{"04F4}
8440 \DeclareUTFCompositeSymbol\{"cyrch"}{"04F5}
8441 \DeclareUTFSymbol\CYRGDSC{"04F6}
8442 \DeclareUTFSymbol\cyrghsc{"04F7}
8443 \DeclareUTFCompositeSymbol\{"CYRERY"}{"04F8}
8444 \DeclareUTFCompositeSymbol\{"cyrery"}{"04F9}
8445 \DeclareUTFSymbol\CYRHHK{"04FC}
8446 \DeclareUTFSymbol\cyrhkh{"04FD}
8447 \DeclareUTFSymbol\sofpasqu{"05C3}
8448 \DeclareUTFSymbol\hebalet{"05D0}
8449 \DeclareUTFSymbol\hebbet{"05D1}
8450 \DeclareUTFSymbol\hebgimel{"05D2}
8451 \DeclareUTFSymbol\hebdaleth{"05D3}
8452 \DeclareUTFSymbol\hebbe{"05D4}
8453 \DeclareUTFSymbol\hebvav{"05D5}
8454 \DeclareUTFSymbol\hebzayin{"05D6}
8455 \DeclareUTFSymbol\hebbet{"05D7}
8456 \DeclareUTFSymbol\hebtet{"05D8}
8457 \DeclareUTFSymbol\hebyod{"05D9}
8458 \DeclareUTFSymbol\hebfinalkaf{"05DA}
8459 \DeclareUTFSymbol\hebkaf{"05DB}
8460 \DeclareUTFSymbol\heblamed{"05DC}
8461 \DeclareUTFSymbol\hebfinalmem{"05DD}
8462 \DeclareUTFSymbol\hebmeme{"05DE}
8463 \DeclareUTFSymbol\hebfinalnun{"05DF}
8464 \DeclareUTFSymbol\hebnun{"05E0}
8465 \DeclareUTFSymbol\hebsamekh{"05E1}
8466 \DeclareUTFSymbol\hebayin{"05E2}
8467 \DeclareUTFSymbol\hebfinalpe{"05E3}
8468 \DeclareUTFSymbol\hebbe{"05E4}
8469 \DeclareUTFSymbol\hebfinaltsadi{"05E5}
8470 \DeclareUTFSymbol\hebttsadi{"05E6}
8471 \DeclareUTFSymbol\hebtsof{"05E7}
8472 \DeclareUTFSymbol\hebreth{"05E8}
8473 \DeclareUTFSymbol\hebshin{"05E9}
8474 \DeclareUTFSymbol\hebtav{"05EA}
8475 \DeclareUTFSymbol\doublevav{"05F0}
8476 \DeclareUTFSymbol\vayod{"05F1}
8477 \DeclareUTFSymbol\doubleyod{"05F2}
8478 \DeclareUTFSymbol\textscd{"1D05}
8479 \DeclareUTFSymbol\textPusc{"1D0B}
8480 \DeclareUTFSymbol\textPusc{"1D0D}
8481 \DeclareUTFSymbol\textPusc{"1D18}

```

```

8482 \DeclareUTFSymbol\textPUrevscr{"1D19}
8483 \DeclareUTFSymbol\textiinferior{"1D62}
8484 \DeclareUTFSymbol\textrinferior{"1D63}
8485 \DeclareUTFSymbol\textuinferior{"1D64}
8486 \DeclareUTFSymbol\textvinferior{"1D65}
8487 \DeclareUTFSymbol\textbetainferior{"1D66}
8488 \DeclareUTFSymbol\textgammainferior{"1D67}
8489 \DeclareUTFSymbol\textrhoinferior{"1D68}
8490 \DeclareUTFSymbol\textphiinferior{"1D69}
8491 \DeclareUTFSymbol\textchiinferior{"1D6A}
8492 \DeclareUTFSymbol\textbarsci{"1D7B}
8493 \DeclareUTFSymbol\textbarp{"1D7D}
8494 \DeclareUTFSymbol\textbarscu{"1D7E}
8495 \DeclareUTFSymbol\textPUrhooka{"1D8F}
8496 \DeclareUTFSymbol\textPUrhook{"1D92}
8497 \DeclareUTFSymbol\textPUrhookepsilon{"1D93}
8498 \DeclareUTFSymbol\textPUrhookopeno{"1D97}
8499 \DeclareUTFCompositeSymbol\textsubbreve{H}{1E2A}
8500 \DeclareUTFCompositeSymbol\textsubbreve{h}{1E2B}
8501 \DeclareUTFCompositeSymbol\.\{\textlong}{1E9B}
8502 \DeclareUTFSymbol\textcompwordmark{"200C}
8503 \DeclareUTFSymbol\textthotfor{"2025}
8504 \DeclareUTFSymbol\textprime{"2032}
8505 \DeclareUTFSymbol\textsecond{"2033}
8506 \DeclareUTFSymbol\textthird{"2034}
8507 \DeclareUTFSymbol\textbackprime{"2035}
8508 \DeclareUTFSymbol\textlefttherefore{"2056}
8509 \DeclareUTFSymbol\textfourth{"2057}
8510 \DeclareUTFSymbol\textdiamonddots{"2058}
8511 \DeclareUTFSymbol\textzerosuperior{"2070}
8512 \DeclareUTFSymbol\textisuperior{"2071}
8513 \DeclareUTFSymbol\textfoursuperior{"2074}
8514 \DeclareUTFSymbol\textfivesuperior{"2075}
8515 \DeclareUTFSymbol\textsixsuperior{"2076}
8516 \DeclareUTFSymbol\textsevensuperior{"2077}
8517 \DeclareUTFSymbol\texteightsuperior{"2078}
8518 \DeclareUTFSymbol\textninesuperior{"2079}
8519 \DeclareUTFSymbol\textplussuperior{"207A}
8520 \DeclareUTFSymbol\textminussuperior{"207B}
8521 \DeclareUTFSymbol\textequalsuperior{"207C}
8522 \DeclareUTFSymbol\textparenleftsuperior{"207D}
8523 \DeclareUTFSymbol\textparenrightsuperior{"207E}
8524 \DeclareUTFSymbol\textnsuperior{"207F}
8525 \DeclareUTFSymbol\textzeroinferior{"2080}
8526 \DeclareUTFSymbol\textoneinferior{"2081}
8527 \DeclareUTFSymbol\texttwoinferior{"2082}
8528 \DeclareUTFSymbol\textthreeinferior{"2083}
8529 \DeclareUTFSymbol\textfourinferior{"2084}
8530 \DeclareUTFSymbol\textfiveinferior{"2085}
8531 \DeclareUTFSymbol\textsixinferior{"2086}
8532 \DeclareUTFSymbol\textseveninferior{"2087}
8533 \DeclareUTFSymbol\texteightinferior{"2088}
8534 \DeclareUTFSymbol\textnineinferior{"2089}
8535 \DeclareUTFSymbol\textplusinferior{"208A}
8536 \DeclareUTFSymbol\textminusinferior{"208B}
8537 \DeclareUTFSymbol\textequalsinferior{"208C}
8538 \DeclareUTFSymbol\textparenleftinferior{"208D}
8539 \DeclareUTFSymbol\textparenrightinferior{"208E}
8540 \DeclareUTFSymbol\textainferior{"2090}
8541 \DeclareUTFSymbol\texteinferior{"2091}
8542 \DeclareUTFSymbol\textoinferior{"2092}
8543 \DeclareUTFSymbol\textxinferior{"2093}
8544 \DeclareUTFSymbol\textschwainferior{"2094}
8545 \DeclareUTFSymbol\textthinferior{"2095}
8546 \DeclareUTFSymbol\textkinferior{"2096}
8547 \DeclareUTFSymbol\textlinferior{"2097}
8548 \DeclareUTFSymbol\textminferior{"2098}

```

```

8549 \DeclareUTFSymbol\textninferior{"2099}
8550 \DeclareUTFSymbol\textpinferior{"209A}
8551 \DeclareUTFSymbol\textsinferior{"209B}
8552 \DeclareUTFSymbol\texttinferior{"209C}
8553 \DeclareUTFSymbol\textpeseta{"20A7}
8554 \DeclareUTFSymbol\textDeleatur{"20B0}
8555 \DeclareUTFSymbol\textguarani{"20B2}
8556 \DeclareUTFSymbol\textslash{"210F}
8557 \DeclareUTFSymbol\textIm{"2111}
8558 \DeclareUTFSymbol\textell{"2113}
8559 \DeclareUTFSymbol\textwp{"2118}
8560 \DeclareUTFSymbol\textRe{"211C}
8561 \DeclareUTFSymbol\textriota{"2129}
8562 \DeclareUTFSymbol\textangstrom{"212B}
8563 \DeclareUTFSymbol\textfax{"213B}
8564 \DeclareUTFSymbol\textinvamp{"214B}
8565 \DeclareUTFSymbol\textoneseventh{"2150}
8566 \DeclareUTFSymbol\textoneninth{"2151}
8567 \DeclareUTFSymbol\textonetenth{"2152}
8568 \DeclareUTFSymbol\textonethird{"2153}
8569 \DeclareUTFSymbol\texttwothirds{"2154}
8570 \DeclareUTFSymbol\textonefifth{"2155}
8571 \DeclareUTFSymbol\texttwofifths{"2156}
8572 \DeclareUTFSymbol\textthreefifths{"2157}
8573 \DeclareUTFSymbol\textfourfifths{"2158}
8574 \DeclareUTFSymbol\textonesixth{"2159}
8575 \DeclareUTFSymbol\textfivesixths{"215A}
8576 \DeclareUTFSymbol\textoneeighth{"215B}
8577 \DeclareUTFSymbol\textthreeeighths{"215C}
8578 \DeclareUTFSymbol\textfiveeighths{"215D}
8579 \DeclareUTFSymbol\textseveneighths{"215E}
8580 \DeclareUTFSymbol\textrevc{"2184}
8581 \DeclareUTFSymbol\textzerothirds{"2189}
8582 \DeclareUTFSymbol\textnleftarrow{"219A}
8583 \DeclareUTFSymbol\textnrightarrow{"219B}
8584 \DeclareUTFSymbol\texttwoheadleftarrow{"219E}
8585 \DeclareUTFSymbol\texttwoheadleftarrow{\textlstrickethru\texttwoheadleftarrow}
8586 \DeclareUTFSymbol\texttwoheaduparrow{"219F}
8587 \DeclareUTFSymbol\texttwoheadrightarrow{"21A0}
8588 \DeclareUTFSymbol\texttwoheadrightarrow{\textlstrickethru\texttwoheadrightarrow}
8589 \DeclareUTFSymbol\texttwoheaddownarrow{"21A1}
8590 \DeclareUTFSymbol\textleftarrowtail{"21A2}
8591 \DeclareUTFSymbol\textrightarrowtail{"21A3}
8592 \DeclareUTFSymbol\textmapsto{"21A6}
8593 \DeclareUTFSymbol\texthookleftarrow{"21A9}
8594 \DeclareUTFSymbol\texthookrightarrow{"21AA}
8595 \DeclareUTFSymbol\textlooparrowleft{"21AB}
8596 \DeclareUTFSymbol\textlooparrowright{"21AC}
8597 \DeclareUTFSymbol\textnleftrightarrow{"21AE}
8598 \DeclareUTFSymbol\textlightning{"21AF}
8599 \DeclareUTFSymbol\textdsh{"21B5}
8600 \DeclareUTFSymbol\textcurvearrowleft{"21B6}
8601 \DeclareUTFSymbol\textcurvearrowright{"21B7}
8602 \DeclareUTFSymbol\textleftharpoonup{"21BC}
8603 \DeclareUTFSymbol\textleftharpoondown{"21BD}
8604 \DeclareUTFSymbol\textupharpoonright{"21BE}
8605 \DeclareUTFSymbol\textupharpoonleft{"21BF}
8606 \DeclareUTFSymbol\texttrightharpoonup{"21C0}
8607 \DeclareUTFSymbol\texttrightharpoondown{"21C1}
8608 \DeclareUTFSymbol\textdownharpoonright{"21C2}
8609 \DeclareUTFSymbol\textdownharpoonleft{"21C3}
8610 \DeclareUTFSymbol\textrightleftarrows{"21C4}
8611 \DeclareUTFSymbol\textupdownarrows{"21C5}
8612 \DeclareUTFSymbol\textleftrigharrows{"21C6}
8613 \DeclareUTFSymbol\textleftleftarrows{"21C7}
8614 \DeclareUTFSymbol\textupuparrows{"21C8}
8615 \DeclareUTFSymbol\texttrightrightarrows{"21C9}

```

```

8616 \DeclareUTFSymbol\textdownarrows{"21CA}
8617 \DeclareUTFSymbol\textleftrightharpoons{"21CB}
8618 \DeclareUTFSymbol\textrightleftharpoons{"21CC}
8619 \DeclareUTFSymbol\textnLeftarrow{"21CD}
8620 \DeclareUTFSymbol\textnLeftarrow{"21CE}
8621 \DeclareUTFSymbol\textnRightarrow{"21CF}
8622 \DeclareUTFSymbol\textLeftarrow{"21D0}
8623 \DeclareUTFSymbol\textUparrow{"21D1}
8624 \DeclareUTFSymbol\textRightarrow{"21D2}
8625 \DeclareUTFSymbol\textDownarrow{"21D3}
8626 \DeclareUTFSymbol\textLeftrightarrow{"21D4}
8627 \DeclareUTFSymbol\textUpdownarrow{"21D5}
8628 \DeclareUTFSymbol\textNarrow{"21D6}
8629 \DeclareUTFSymbol\textNearrow{"21D7}
8630 \DeclareUTFSymbol\textSearrow{"21D8}
8631 \DeclareUTFSymbol\textSwarrow{"21D9}
8632 \DeclareUTFSymbol\textLleftarrow{"21DA}
8633 \DeclareUTFSymbol\textRrightarrow{"21DB}
8634 \DeclareUTFSymbol\textleftsquigarrow{"21DC}
8635 \DeclareUTFSymbol\textrightsquigarrow{"21DD}
8636 \DeclareUTFSymbol\textdashleftarrow{"21E0}
8637 \DeclareUTFSymbol\textdasheduparrow{"21E1}
8638 \DeclareUTFSymbol\textdashrightarrow{"21E2}
8639 \DeclareUTFSymbol\textdasheddownarrow{"21E3}
8640 \DeclareUTFSymbol\textpointer{"21E8}
8641 \DeclareUTFSymbol\textdownuparrows{"21F5}
8642 \DeclareUTFSymbol\textleftarrowtriangle{"21FD}
8643 \DeclareUTFSymbol\textrightarrowtriangle{"21FE}
8644 \DeclareUTFSymbol\textleftrightharpoontriangle{"21FF}
8645 \DeclareUTFSymbol\textforall{"2200}
8646 \DeclareUTFSymbol\textcomplement{"2201}
8647 \DeclareUTFSymbol\textpartial{"2202}
8648 \DeclareUTFSymbol\textexists{"2203}
8649 \DeclareUTFSymbol\textnexists{"2204}
8650 \DeclareUTFSymbol\textemptyset{"2205}
8651 \DeclareUTFSymbol\texttriangle{"2206}
8652 \DeclareUTFSymbol\textnabla{"2207}
8653 \DeclareUTFSymbol\textin{"2208}
8654 \DeclareUTFSymbol\textnotin{"2209}
8655 \DeclareUTFSymbol\textsmallin{"220A}
8656 \DeclareUTFSymbol\textni{"220B}
8657 \DeclareUTFSymbol\textnotowner{"220C}
8658 \DeclareUTFSymbol\textsmallowns{"220D}
8659 \DeclareUTFSymbol\textprod{"220F}
8660 \DeclareUTFSymbol\textamalg{"2210}
8661 \DeclareUTFSymbol\textsum{"2211}
8662 \DeclareUTFSymbol\textmp{"2213}
8663 \DeclareUTFSymbol\textdotplus{"2214}
8664 \DeclareUTFSymbol\textDivides{"2215}
8665 \DeclareUTFSymbol\textsetminus{"2216}
8666 \DeclareUTFSymbol\textast{"2217}
8667 \DeclareUTFSymbol\textcirc{"2218}
8668 \DeclareUTFSymbol\textbulletoperator{"2219}
8669 \DeclareUTFSymbol\textpropto{"221D}
8670 \DeclareUTFSymbol\textinfty{"221E}
8671 \DeclareUTFSymbol\textangle{"2220}
8672 \DeclareUTFSymbol\textmeasuredangle{"2221}
8673 \DeclareUTFSymbol\textsphericalangle{"2222}
8674 \DeclareUTFSymbol\textmid{"2223}
8675 \DeclareUTFSymbol\textnmid{"2224}
8676 \DeclareUTFSymbol\textparallel{"2225}
8677 \DeclareUTFSymbol\textnparallel{"2226}
8678 \DeclareUTFSymbol\textwedge{"2227}
8679 \DeclareUTFCommand\textowedge{\textcircled\textwedge}
8680 \DeclareUTFSymbol\textvee{"2228}
8681 \DeclareUTFCommand\textovee{\textcircled\textvee}
8682 \DeclareUTFSymbol\textcap{"2229}

```

```

8683 \DeclareUTFSymbol\textcup{"222A}
8684 \DeclareUTFSymbol\textint{"222B}
8685 \DeclareUTFSymbol\textiint{"222C}
8686 \DeclareUTFSymbol\textiiint{"222D}
8687 \DeclareUTFSymbol\textoint{"222E}
8688 \DeclareUTFSymbol\textoint{"222F}
8689 \DeclareUTFSymbol\textointclockwise{"2232}
8690 \DeclareUTFSymbol\textointctrclockwise{"2233}
8691 \DeclareUTFSymbol\texttherefore{"2234}
8692 \DeclareUTFSymbol\textbecause{"2235}
8693 \DeclareUTFSymbol\textvdotdot{"2236}
8694 \DeclareUTFSymbol\textsquaredots{"2237}
8695 \DeclareUTFSymbol\textdotminus{"2238}
8696 \DeclareUTFSymbol\texteqcolon{"2239}
8697 \DeclareUTFSymbol\textsim{"223C}
8698 \DeclareUTFSymbol\textbacksim{"223D}
8699 \DeclareUTFCommand\textnbacksim{\textlstrikethru\textbacksim}
8700 \DeclareUTFSymbol\textwr{"2240}
8701 \DeclareUTFSymbol\textnsim{"2241}
8702 \DeclareUTFSymbol\texteqsim{"2242}
8703 \DeclareUTFCommand\textneqsim{\textlstrikethru\texteqsim}
8704 \DeclareUTFSymbol\textsimeq{"2243}
8705 \DeclareUTFSymbol\textnsimeq{"2244}
8706 \DeclareUTFSymbol\textcong{"2245}
8707 \DeclareUTFSymbol\textncong{"2247}
8708 \DeclareUTFSymbol\textapprox{"2248}
8709 \DeclareUTFSymbol\textnapprox{"2249}
8710 \DeclareUTFSymbol\textapproxeq{"224A}
8711 \DeclareUTFCommand\textnapproxeq{\textlstrikethru\textapproxeq}
8712 \DeclareUTFSymbol\texttriplesim{"224B}
8713 \DeclareUTFCommand\textntriplesim{\textlstrikethru\texttriplesim}
8714 \DeclareUTFSymbol\textbackcong{"224C}
8715 \DeclareUTFCommand\textnbackcong{\textlstrikethru\textbackcong}
8716 \DeclareUTFSymbol\textasympt{"224D}
8717 \DeclareUTFCommand\textnasympt{\textlstrikethru\textasympt}
8718 \DeclareUTFSymbol\textBumpeq{"224E}
8719 \DeclareUTFCommand\textnBumpeq{\textlstrikethru\textBumpeq}
8720 \DeclareUTFSymbol\textbumpeq{"224F}
8721 \DeclareUTFCommand\textnbumpeq{\textlstrikethru\textbumpeq}
8722 \DeclareUTFSymbol\textdoteq{"2250}
8723 \DeclareUTFCommand\textndoteq{\textlstrikethru\textdoteq}
8724 \DeclareUTFSymbol\textdoteqdot{"2251}
8725 \DeclareUTFCommand\textnDoteq{\textlstrikethru\textdoteqdot}
8726 \DeclareUTFSymbol\textfallingdoteq{"2252}
8727 \DeclareUTFCommand\textnfallingdoteq{\textlstrikethru\textfallingdoteq}
8728 \DeclareUTFSymbol\textrisingdoteq{"2253}
8729 \DeclareUTFCommand\textnrisingdoteq{\textlstrikethru\textrisingdoteq}
8730 \DeclareUTFSymbol\textcolonequals{"2254}
8731 \DeclareUTFSymbol\textequalscolon{"2255}
8732 \DeclareUTFSymbol\texteqcirc{"2256}
8733 \DeclareUTFCommand\textneqcirc{\textlstrikethru\texteqcirc}
8734 \DeclareUTFSymbol\textcirceq{"2257}
8735 \DeclareUTFCommand\textncirceq{\textlstrikethru\textcirceq}
8736 \DeclareUTFSymbol\texthateq{"2259}
8737 \DeclareUTFCommand\textnhateq{\textlstrikethru\texthateq}
8738 \DeclareUTFSymbol\texttriangleeq{"225C}
8739 \DeclareUTFSymbol\textneq{"2260}
8740 \DeclareUTFSymbol\textneq{"2260}
8741 \DeclareUTFSymbol\textequiv{"2261}
8742 \DeclareUTFSymbol\textnequiv{"2262}
8743 \DeclareUTFSymbol\textleq{"2264}
8744 \DeclareUTFSymbol\textleq{"2264}
8745 \DeclareUTFSymbol\textgeq{"2265}
8746 \DeclareUTFSymbol\textgeq{"2265}
8747 \DeclareUTFSymbol\textleqq{"2266}
8748 \DeclareUTFCommand\textnleqq{\textlstrikethru\textleqq}
8749 \DeclareUTFSymbol\textgeqq{"2267}

```

```

8750 \DeclareUTFCommand\textngeqq{\textlstrikethru\textgeqq}
8751 \DeclareUTFSymbol\textlneqq{"2268}
8752 \DeclareUTFSymbol\textgneqq{"2269}
8753 \DeclareUTFSymbol\textll{"226A}
8754 \DeclareUTFCommand\textnll{\textlstrikethru\textll}
8755 \DeclareUTFSymbol\textgg{"226B}
8756 \DeclareUTFCommand\textngg{\textlstrikethru\textgg}
8757 \DeclareUTFSymbol\textbetween{"226C}
8758 \DeclareUTFSymbol\textnless{"226E}
8759 \DeclareUTFSymbol\textngtr{"226F}
8760 \DeclareUTFSymbol\textnleq{"2270}
8761 \DeclareUTFSymbol\textngeq{"2271}
8762 \DeclareUTFSymbol\textlessssim{"2272}
8763 \DeclareUTFSymbol\textgtrsim{"2273}
8764 \DeclareUTFSymbol\textnlessssim{"2274}
8765 \DeclareUTFSymbol\textngtrsim{"2275}
8766 \DeclareUTFSymbol\textlessgtr{"2276}
8767 \DeclareUTFSymbol\textgtrless{"2277}
8768 \DeclareUTFSymbol\textngtrless{"2278}
8769 \DeclareUTFSymbol\textnlessgtr{"2279}
8770 \DeclareUTFSymbol\textprec{"227A}
8771 \DeclareUTFSymbol\textsucc{"227B}
8772 \DeclareUTFSymbol\textpreccurlyeq{"227C}
8773 \DeclareUTFSymbol\textsucceedcurlyeq{"227D}
8774 \DeclareUTFSymbol\textprecsim{"227E}
8775 \DeclareUTFCommand\textnprecsim{\textlstrikethru\textprecsim}
8776 \DeclareUTFSymbol\textsuccsim{"227F}
8777 \DeclareUTFCommand\textnsuccsim{\textlstrikethru\textsuccsim}
8778 \DeclareUTFSymbol\textnprec{"2280}
8779 \DeclareUTFSymbol\textnsucc{"2281}
8780 \DeclareUTFSymbol\textsubset{"2282}
8781 \DeclareUTFSymbol\textsupset{"2283}
8782 \DeclareUTFSymbol\textnsubset{"2284}
8783 \DeclareUTFSymbol\textnsupset{"2285}
8784 \DeclareUTFSymbol\textsubseteq{"2286}
8785 \DeclareUTFSymbol\textsupseteq{"2287}
8786 \DeclareUTFSymbol\textnsubseteq{"2288}
8787 \DeclareUTFSymbol\textnsupseteq{"2289}
8788 \DeclareUTFSymbol\textsubsetneq{"228A}
8789 \DeclareUTFSymbol\textsupsetneq{"228B}
8790 \DeclareUTFSymbol\textcupdot{"228D}
8791 \DeclareUTFSymbol\textcupplus{"228E}
8792 \DeclareUTFSymbol\textsqsubset{"228F}
8793 \DeclareUTFCommand\textnsqsubset{\textlstrikethru\textsqsubset}
8794 \DeclareUTFSymbol\textsqsupset{"2290}
8795 \DeclareUTFCommand\textnsqsupset{\textlstrikethru\textsqsupset}
8796 \DeclareUTFSymbol\textsqsubseq{"2291}
8797 \DeclareUTFCommand\textnsqsubseq{\textlstrikethru\textsqsubseq}
8798 \DeclareUTFSymbol\textsqsupseteq{"2292}
8799 \DeclareUTFCommand\textnsqsupseteq{\textlstrikethru\textsqsupseteq}
8800 \DeclareUTFSymbol\textsqcap{"2293}
8801 \DeclareUTFSymbol\textsqcup{"2294}
8802 \DeclareUTFSymbol\textoplus{"2295}
8803 \DeclareUTFSymbol\textominus{"2296}
8804 \DeclareUTFSymbol\textotimes{"2297}
8805 \DeclareUTFSymbol\textoslash{"2298}
8806 \DeclareUTFSymbol\textodot{"2299}
8807 \DeclareUTFSymbol\textcircledcirc{"229A}
8808 \DeclareUTFSymbol\textcircledast{"229B}
8809 \DeclareUTFSymbol\textcircledash{"229D}
8810 \DeclareUTFSymbol\textboxplus{"229E}
8811 \DeclareUTFSymbol\textboxminus{"229F}
8812 \DeclareUTFSymbol\textboxtimes{"22A0}
8813 \DeclareUTFSymbol\textboxdot{"22A1}
8814 \DeclareUTFSymbol\textvdash{"22A2}
8815 \DeclareUTFSymbol\textdashv{"22A3}
8816 \DeclareUTFCommand\textndashv{\textlstrikethru\textdashv}

```

```

8817 \DeclareUTFSymbol\texttop{"22A4}
8818 \DeclareUTFCommand\textndownvdash{\textlstrikethru\texttop}
8819 \DeclareUTFSymbol\textbot{"22A5}
8820 \DeclareUTFCommand\textnupvdash{\textlstrikethru\textbot}
8821 \DeclareUTFSymbol\textvDash{"22A8}
8822 \DeclareUTFSymbol\textVdash{"22A9}
8823 \DeclareUTFSymbol\textVdash{"22AA}
8824 \DeclareUTFCommand\textnVdash{\textlstrikethru\textVdash}
8825 \DeclareUTFSymbol\textVDash{"22AB}
8826 \DeclareUTFSymbol\textnvdash{"22AC}
8827 \DeclareUTFSymbol\textnvDash{"22AD}
8828 \DeclareUTFSymbol\textnVdash{"22AE}
8829 \DeclareUTFSymbol\textnVDash{"22AF}
8830 \DeclareUTFSymbol\textlhd{"22B2}
8831 \DeclareUTFSymbol\textrhd{"22B3}
8832 \DeclareUTFSymbol\textunlhd{"22B4}
8833 \DeclareUTFSymbol\textunrhd{"22B5}
8834 \DeclareUTFSymbol\textmultimapdotbothA{"22B6}
8835 \DeclareUTFSymbol\textmultimapdotbothB{"22B7}
8836 \DeclareUTFSymbol\textmultimap{"22B8}
8837 \DeclareUTFSymbol\textveebar{"22BB}
8838 \DeclareUTFSymbol\textbarwedge{"22BC}
8839 \DeclareUTFSymbol\textstar{"22C6}
8840 \DeclareUTFSymbol\textdivideontimes{"22C7}
8841 \DeclareUTFSymbol\textbowtie{"22C8}
8842 \DeclareUTFSymbol\textltimes{"22C9}
8843 \DeclareUTFSymbol\textrtimes{"22CA}
8844 \DeclareUTFSymbol\textleftthreetimes{"22CB}
8845 \DeclareUTFSymbol\textrightthreetimes{"22CC}
8846 \DeclareUTFSymbol\textbacksimeq{"22CD}
8847 \DeclareUTFCommand\textnbacksimeq{\textlstrikethru\textbacksimeq}
8848 \DeclareUTFSymbol\textcurlyvee{"22CE}
8849 \DeclareUTFSymbol\textcurlywedge{"22CF}
8850 \DeclareUTFSymbol\textSubset{"22D0}
8851 \DeclareUTFCommand\textnSubset{\textlstrikethru\textSubset}
8852 \DeclareUTFSymbol\textSupset{"22D1}
8853 \DeclareUTFCommand\textnSupset{\textlstrikethru\textSupset}
8854 \DeclareUTFSymbol\textCap{"22D2}
8855 \DeclareUTFSymbol\textCup{"22D3}
8856 \DeclareUTFSymbol\textpitchfork{"22D4}
8857 \DeclareUTFSymbol\textlessdot{"22D6}
8858 \DeclareUTFSymbol\textgtrdot{"22D7}
8859 \DeclareUTFSymbol\textlll{"22D8}
8860 \DeclareUTFSymbol\textggg{"22D9}
8861 \DeclareUTFSymbol\textlesseqgtr{"22DA}
8862 \DeclareUTFSymbol\textgtreqless{"22DB}
8863 \DeclareUTFSymbol\textcurlyeqprec{"22DE}
8864 \DeclareUTFCommand\textncurlyeqprec{\textlstrikethru\textcurlyeqprec}
8865 \DeclareUTFSymbol\textcurlyeqsucc{"22DF}
8866 \DeclareUTFCommand\textncurlyeqsucc{\textlstrikethru\textcurlyeqsucc}
8867 \DeclareUTFSymbol\textnpreccurlyeq{"22E0}
8868 \DeclareUTFSymbol\textnsucccurlyeq{"22E1}
8869 \DeclareUTFSymbol\textnqsubsetq{"22E2}
8870 \DeclareUTFSymbol\textnqsupsetq{"22E3}
8871 \DeclareUTFSymbol\textsqsubsetneq{"22E4}
8872 \DeclareUTFSymbol\textsqsupsetneq{"22E5}
8873 \DeclareUTFSymbol\textlnsim{"22E6}
8874 \DeclareUTFSymbol\textgnsim{"22E7}
8875 \DeclareUTFSymbol\textprecnsim{"22E8}
8876 \DeclareUTFSymbol\textsuccnsim{"22E9}
8877 \DeclareUTFSymbol\textntriangleleft{"22EA}
8878 \DeclareUTFSymbol\textntriangleright{"22EB}
8879 \DeclareUTFSymbol\textntrianglelefteq{"22EC}
8880 \DeclareUTFSymbol\textntrianglerighteq{"22ED}
8881 \DeclareUTFSymbol\textvdots{"22EE}
8882 \DeclareUTFSymbol\textcdots{"22EF}
8883 \DeclareUTFSymbol\textudots{"22F0}

```

```
8884 \DeclareUTFSymbol\textddots{"22F1}
8885 \DeclareUTFSymbol\textbarin{"22F6}
8886 \DeclareUTFSymbol\textdiameter{"2300}
8887 \DeclareUTFSymbol\textbackneg{"2310}
8888 \DeclareUTFSymbol\textwasylounge{"2311}
8889 \DeclareUTFSymbol\textinvbackneg{"2319}
8890 \DeclareUTFSymbol\textclock{"231A}
8891 \DeclareUTFSymbol\textulcorner{"231C}
8892 \DeclareUTFSymbol\texturcorner{"231D}
8893 \DeclareUTFSymbol\textllcorner{"231E}
8894 \DeclareUTFSymbol\textlrcorner{"231F}
8895 \DeclareUTFSymbol\textfrown{"2322}
8896 \DeclareUTFSymbol\textsmile{"2323}
8897 \DeclareUTFSymbol\textKeyboard{"2328}
8898 \DeclareUTFSymbol\textlangle{"2329}
8899 \DeclareUTFSymbol\textrangle{"232A}
8900 \DeclareUTFSymbol\textAPLinvert{"2339}
8901 \DeclareUTFSymbol\textTumbler{"233C}
8902 \DeclareUTFSymbol\textstmaryrdbar{"233D}
8903 \DeclareUTFSymbol\textnotslash{"233F}
8904 \DeclareUTFSymbol\textnotbackslash{"2340}
8905 \DeclareUTFSymbol\textboxbackslash{"2342}
8906 \DeclareUTFSymbol\textAPLleftarrowbox{"2347}
8907 \DeclareUTFSymbol\textAPLrightarrowbox{"2348}
8908 \DeclareUTFSymbol\textAPLuparrowbox{"2350}
8909 \DeclareUTFSymbol\textAPLdownarrowbox{"2357}
8910 \DeclareUTFSymbol\textAPLinput{"235E}
8911 \DeclareUTFSymbol\textRequest{"2370}
8912 \DeclareUTFSymbol\textBeam{"2393}
8913 \DeclareUTFSymbol\texthexagon{"2394}
8914 \DeclareUTFSymbol\textAPLbox{"2395}
8915 \DeclareUTFSymbol\textForwardToIndex{"23ED}
8916 \DeclareUTFSymbol\textRewindToIndex{"23EE}
8917 \DeclareUTFSymbol\textbbslash{"244A}
8918 \DeclareUTFSymbol\textCircledA{"24B6}
8919 \DeclareUTFSymbol\textCleaningF{"24BB}
8920 \DeclareUTFSymbol\textCleaningFF{\b\textCleaningF}
8921 \DeclareUTFSymbol\textCleaningP{"24C5}
8922 \DeclareUTFSymbol\textCleaningPP{\b\textCleaningP}
8923 \DeclareUTFSymbol\textCuttingLine{"2504}
8924 \DeclareUTFSymbol\textUParrow{"25B2}
8925 \DeclareUTFSymbol\textbigtriangleup{"25B3}
8926 \DeclareUTFSymbol\textForward{"25B6}
8927 \DeclareUTFSymbol\texttriangleright{"25B7}
8928 \DeclareUTFSymbol\textRHD{"25BA}
8929 \DeclareUTFSymbol\textDOWNarrow{"25BC}
8930 \DeclareUTFSymbol\textbigtriangledown{"25BD}
8931 \DeclareUTFSymbol\textRewind{"25C0}
8932 \DeclareUTFSymbol\texttriangleleft{"25C1}
8933 \DeclareUTFSymbol\textLHD{"25C4}
8934 \DeclareUTFSymbol\textdiamond{"25C7}
8935 \DeclareUTFSymbol\textlozenge{"25CA}
8936 \DeclareUTFSymbol\textLEFTCIRCLE{"25D6}
8937 \DeclareUTFSymbol\textRIGHTCIRCLE{"25D7}
8938 \DeclareUTFSymbol\textboxbar{"25EB}
8939 \DeclareUTFSymbol\textCloud{"2601}
8940 \DeclareUTFSymbol\textFiveStar{"2605}
8941 \DeclareUTFSymbol\textFiveStarOpen{"2606}
8942 \DeclareUTFSymbol\textPhone{"260E}
8943 \DeclareUTFSymbol\textboxempty{"2610}
8944 \DeclareUTFSymbol\textCheckedbox{"2611}
8945 \DeclareUTFSymbol\textCrossedbox{"2612}
8946 \DeclareUTFSymbol\textCoffeecup{"2615}
8947 \DeclareUTFSymbol\textHandCuffLeft{"261A}
8948 \DeclareUTFSymbol\textHandCuffRight{"261B}
8949 \DeclareUTFSymbol\textHandLeft{"261C}
8950 \DeclareUTFSymbol\textHandRight{"261E}
```

```
8951 \DeclareUTFSymbol\textRadioactivity{"2622}  
8952 \DeclareUTFSymbol\textBiohazard{"2623}  
8953 \DeclareUTFSymbol\textAnkh{"2625}  
8954 \DeclareUTFSymbol\textYinYang{"262F}  
8955 \DeclareUTFSymbol\textfrownie{"2639}  
8956 \DeclareUTFSymbol\textsmiley{"263A}  
8957 \DeclareUTFSymbol\textblacksmiley{"263B}  
8958 \DeclareUTFSymbol\textsun{"263C}  
8959 \DeclareUTFSymbol\textleftmoon{"263D}  
8960 \DeclareUTFSymbol\textrightmoon{"263E}  
8961 \DeclareUTFSymbol\textmercury{"263F}  
8962 \DeclareUTFSymbol\textPUfemale{"2640}  
8963 \DeclareUTFSymbol\textearth{"2641}  
8964 \DeclareUTFSymbol\textmale{"2642}  
8965 \DeclareUTFSymbol\textjupiter{"2643}  
8966 \DeclareUTFSymbol\textsaturn{"2644}  
8967 \DeclareUTFSymbol\texturanus{"2645}  
8968 \DeclareUTFSymbol\textneptune{"2646}  
8969 \DeclareUTFSymbol\textpluto{"2647}  
8970 \DeclareUTFSymbol\textaries{"2648}  
8971 \DeclareUTFSymbol\texttaurus{"2649}  
8972 \DeclareUTFSymbol\textgemini{"264A}  
8973 \DeclareUTFSymbol\textcancer{"264B}  
8974 \DeclareUTFSymbol\textleo{"264C}  
8975 \DeclareUTFSymbol\textvirgo{"264D}  
8976 \DeclareUTFSymbol\textlibra{"264E}  
8977 \DeclareUTFSymbol\textscorpio{"264F}  
8978 \DeclareUTFSymbol\textsagittarius{"2650}  
8979 \DeclareUTFSymbol\textcapricornus{"2651}  
8980 \DeclareUTFSymbol\textaquarius{"2652}  
8981 \DeclareUTFSymbol\textpisces{"2653}  
8982 \DeclareUTFSymbol\textspadesuitblack{"2660}  
8983 \DeclareUTFSymbol\textheartsuitwhite{"2661}  
8984 \DeclareUTFSymbol\textdiamondsuitwhite{"2662}  
8985 \DeclareUTFSymbol\textclubsuitblack{"2663}  
8986 \DeclareUTFSymbol\textspadesuitwhite{"2664}  
8987 \DeclareUTFSymbol\textheartsuitblack{"2665}  
8988 \DeclareUTFSymbol\textdiamondsuitblack{"2666}  
8989 \DeclareUTFSymbol\textclubsuitwhite{"2667}  
8990 \DeclareUTFSymbol\textquarternote{"2669}  
8991 \DeclareUTFSymbol\texttwonotes{"266B}  
8992 \DeclareUTFSymbol\textsixteenthnote{"266C}  
8993 \DeclareUTFSymbol\textflat{"266D}  
8994 \DeclareUTFSymbol\textnatural{"266E}  
8995 \DeclareUTFSymbol\textsharp{"266F}  
8996 \DeclareUTFSymbol\textrecycle{"2672}  
8997 \DeclareUTFSymbol\textWheelchair{"267F}  
8998 \DeclareUTFSymbol\textFlag{"2691}  
8999 \DeclareUTFSymbol\textMineSign{"2692}  
9000 \DeclareUTFSymbol\textdsmilitary{"2694}  
9001 \DeclareUTFSymbol\textdsmedical{"2695}  
9002 \DeclareUTFSymbol\textdsjuridical{"2696}  
9003 \DeclareUTFSymbol\textdschemical{"2697}  
9004 \DeclareUTFSymbol\textdsbiological{"2698}  
9005 \DeclareUTFSymbol\textdscommercial{"269A}  
9006 \DeclareUTFSymbol\textmanstar{"269D}  
9007 \DeclareUTFSymbol\textdanger{"26A0}  
9008 \DeclareUTFSymbol\textFemaleFemale{"26A2}  
9009 \DeclareUTFSymbol\textMaleMale{"26A3}  
9010 \DeclareUTFSymbol\textFemaleMale{"26A4}  
9011 \DeclareUTFSymbol\textHermaphrodite{"26A5}  
9012 \DeclareUTFSymbol\textNeutral{"26AA}  
9013 \DeclareUTFSymbol\textPUuncrfemale{"26B2}  
9014 \DeclareUTFSymbol\texthexstar{"26B9}  
9015 \DeclareUTFSymbol\textSoccerBall{"26BD}  
9016 \DeclareUTFSymbol\textSunCloud{"26C5}  
9017 \DeclareUTFSymbol\textRain{"26C6}
```

```

9018 \DeclareUTFSymbol\textnoway{"26D4}
9019 \DeclareUTFSymbol\textMountain{"26F0}
9020 \DeclareUTFSymbol\textTent{"26FA}
9021 \DeclareUTFSymbol\textScissorRightBrokenBottom{"2701}
9022 \DeclareUTFSymbol\textScissorRight{"2702}
9023 \DeclareUTFSymbol\textScissorRightBrokenTop{"2703}
9024 \DeclareUTFSymbol\textScissorHollowRight{"2704}
9025 \DeclareUTFSymbol\textPhoneHandset{"2706}
9026 \DeclareUTFSymbol\textTape{"2707}
9027 \DeclareUTFSymbol\textPlane{"2708}
9028 \DeclareUTFSymbol\textEnvelope{"2709}
9029 \DeclareUTFSymbol\textPeace{"270C}
9030 \DeclareUTFSymbol\textWritingHand{"270D}
9031 \DeclareUTFSymbol\textPencilRightDown{"270E}
9032 \DeclareUTFSymbol\textPencilRight{"270F}
9033 \DeclareUTFSymbol\textPencilRightUp{"2710}
9034 \DeclareUTFSymbol\textNibRight{"2711}
9035 \DeclareUTFSymbol\textNibSolidRight{"2712}
9036 \DeclareUTFSymbol\textCheckmark{"2713}
9037 \DeclareUTFSymbol\textCheckmarkBold{"2714}
9038 \DeclareUTFSymbol\textXSolid{"2715}
9039 \DeclareUTFSymbol\textXSolidBold{"2716}
9040 \DeclareUTFSymbol\textXSolidBrush{"2717}
9041 \DeclareUTFSymbol\textPlusOutline{"2719}
9042 \DeclareUTFSymbol\textPlus{"271A}
9043 \DeclareUTFSymbol\textPlusThinCenterOpen{"271B}
9044 \DeclareUTFSymbol\textPlusCenterOpen{"271C}
9045 \DeclareUTFSymbol\textCross{"271D}
9046 \DeclareUTFSymbol\textCrossOpenShadow{"271E}
9047 \DeclareUTFSymbol\textCrossOutline{"271F}
9048 \DeclareUTFSymbol\textCrossMaltese{"2720}
9049 \DeclareUTFSymbol\textDavidStar{"2721}
9050 \DeclareUTFSymbol\textFourAsterisk{"2722}
9051 \DeclareUTFSymbol\textJackStar{"2723}
9052 \DeclareUTFSymbol\textJackStarBold{"2724}
9053 \DeclareUTFSymbol\textClowerTips{"2725}
9054 \DeclareUTFSymbol\textFourStar{"2726}
9055 \DeclareUTFSymbol\textFourStarOpen{"2727}
9056 \DeclareUTFSymbol\textFiveStarOpenCircled{"272A}
9057 \DeclareUTFSymbol\textFiveStarCenterOpen{"272B}
9058 \DeclareUTFSymbol\textFiveStarOpenDotted{"272C}
9059 \DeclareUTFSymbol\textFiveStarOutline{"272D}
9060 \DeclareUTFSymbol\textFiveStarOutlineHeavy{"272E}
9061 \DeclareUTFSymbol\textFiveStarConvex{"272F}
9062 \DeclareUTFSymbol\textFiveStarShadow{"2730}
9063 \DeclareUTFSymbol\textAsteriskBold{"2731}
9064 \DeclareUTFSymbol\textAsteriskCenterOpen{"2732}
9065 \DeclareUTFSymbol\textEightStarTaper{"2734}
9066 \DeclareUTFSymbol\textEightStarConvex{"2735}
9067 \DeclareUTFSymbol\textSixStar{"2736}
9068 \DeclareUTFSymbol\textEightStar{"2737}
9069 \DeclareUTFSymbol\textEightStarBold{"2738}
9070 \DeclareUTFSymbol\textTwelveStar{"2739}
9071 \DeclareUTFSymbol\textSixteenStarLight{"273A}
9072 \DeclareUTFSymbol\textSixFlowerPetalRemoved{"273B}
9073 \DeclareUTFSymbol\textSixFlowerOpenCenter{"273C}
9074 \DeclareUTFSymbol\textAsterisk{"273D}
9075 \DeclareUTFSymbol\textSixFlowerAlternate{"273E}
9076 \DeclareUTFSymbol\textFiveFlowerPetal{"273F}
9077 \DeclareUTFSymbol\textFiveFlowerOpen{"2740}
9078 \DeclareUTFSymbol\textEightFlowerPetal{"2741}
9079 \DeclareUTFSymbol\textSunshineOpenCircled{"2742}
9080 \DeclareUTFSymbol\textSixFlowerAltPetal{"2743}
9081 \DeclareUTFSymbol\textSnowflakeChevron{"2744}
9082 \DeclareUTFSymbol\textSnowflake{"2745}
9083 \DeclareUTFSymbol\textSnowflakeChevronBold{"2746}
9084 \DeclareUTFSymbol\textSparkle{"2747}

```

```

9085 \DeclareUTFSymbol\textSparkleBold{"2748}
9086 \DeclareUTFSymbol\textAsteriskRoundedEnds{"2749}
9087 \DeclareUTFSymbol\textEightFlowerPetalRemoved{"274A}
9088 \DeclareUTFSymbol\textEightAsterisk{"274B}
9089 \DeclareUTFSymbol\textCircleShadow{"274D}
9090 \DeclareUTFSymbol\textSquareShadowBottomRight{"274F}
9091 \DeclareUTFSymbol\textSquareTopRight{"2750}
9092 \DeclareUTFSymbol\textSquareCastShadowBottomRight{"2751}
9093 \DeclareUTFSymbol\textSquareCastShadowTopRight{"2752}
9094 \DeclareUTFSymbol\textDiamondSolid{"2756}
9095 \DeclareUTFSymbol\textRectangleThin{"2758}
9096 \DeclareUTFSymbol\textRectangle{"2759}
9097 \DeclareUTFSymbol\textRectangleBold{"275A}
9098 \DeclareUTFSymbol\textperp{"27C2}
9099 \DeclareUTFCommand\textnotperp{\textlstrikethru\textperp}
9100 \DeclareUTFSymbol\textveedot{"27C7}
9101 \DeclareUTFSymbol\textwedgedot{"27D1}
9102 \DeclareUTFSymbol\textleftspoon{"27DC}
9103 \DeclareUTFSymbol\textlbrackdbl{"27E6}
9104 \DeclareUTFSymbol\textlbrackdbl{"27E7}
9105 \DeclareUTFSymbol\textcirclearrowleft{"27F2}
9106 \DeclareUTFSymbol\textcirclearrowright{"27F3}
9107 \DeclareUTFSymbol\textlongleftarrow{"27F5}
9108 \DeclareUTFSymbol\textlongrightarrow{"27F6}
9109 \DeclareUTFSymbol\textlongleftrightarrow{"27F7}
9110 \DeclareUTFSymbol\textLongleftarrow{"27F8}
9111 \DeclareUTFSymbol\textLongrightarrow{"27F9}
9112 \DeclareUTFSymbol\textLongleftrightarrow{"27FA}
9113 \DeclareUTFSymbol\textlongmapsto{"27FC}
9114 \DeclareUTFSymbol\textLongmapsfrom{"27FD}
9115 \DeclareUTFSymbol\textLongmapsto{"27FE}
9116 \DeclareUTFSymbol\textnwsearrow{"2921}
9117 \DeclareUTFSymbol\textneswarrow{"2922}
9118 \DeclareUTFSymbol\textlhooknwarrow{"2923}
9119 \DeclareUTFSymbol\textlhooknwarrow{"2924}
9120 \DeclareUTFSymbol\textlhooksearrow{"2925}
9121 \DeclareUTFSymbol\textlhooksearrow{"2926}
9122 \DeclareUTFSymbol\textleadsto{"2933}
9123 \DeclareUTFSymbol\textrcurvearrowne{"2934}
9124 \DeclareUTFSymbol\textlcurvearrowse{"2935}
9125 \DeclareUTFSymbol\textlcurvearrowsw{"2936}
9126 \DeclareUTFSymbol\textrcurvearrowse{"2937}
9127 \DeclareUTFSymbol\textlcurvearrowdown{"2938}
9128 \DeclareUTFSymbol\textrcurvearrowdown{"2939}
9129 \DeclareUTFSymbol\textrcurvearrowleft{"293A}
9130 \DeclareUTFSymbol\textrcurvearrowright{"293B}
9131 \DeclareUTFSymbol\textleftrightharpoon{"294A}
9132 \DeclareUTFSymbol\textrightleftharpoon{"294B}
9133 \DeclareUTFSymbol\textupdownharpoonrightleft{"294C}
9134 \DeclareUTFSymbol\textupdownharpoonleftrigh{"294D}
9135 \DeclareUTFSymbol\textleftleftharpoons{"2962}
9136 \DeclareUTFSymbol\textupupharpoons{"2963}
9137 \DeclareUTFSymbol\texttrightrightarpoons{"2964}
9138 \DeclareUTFSymbol\textdowndownharpoons{"2965}
9139 \DeclareUTFSymbol\textleftbarharpoon{"296A}
9140 \DeclareUTFSymbol\textbarleftharpoon{"296B}
9141 \DeclareUTFSymbol\textrightbarharpoon{"296C}
9142 \DeclareUTFSymbol\textbarriharpoon{"296D}
9143 \DeclareUTFSymbol\textupdownharpoons{"296E}
9144 \DeclareUTFSymbol\textdownupharpoons{"296F}
9145 \DeclareUTFSymbol\textllparenthesis{"2987}
9146 \DeclareUTFSymbol\textrrparenthesis{"2988}
9147 \DeclareUTFSymbol\textinvdiameter{"29B0}
9148 \DeclareUTFSymbol\textobar{"29B6}
9149 \DeclareUTFSymbol\textobslash{"29B8}
9150 \DeclareUTFSymbol\textobot{"29BA}
9151 \DeclareUTFSymbol\textNoChemicalCleaning{"29BB}

```

```

9152 \DeclareUTFSymbol\textolessthan{"29C0}
9153 \DeclareUTFSymbol\textogreaterthan{"29C1}
9154 \DeclareUTFSymbol\textboxslash{"29C4}
9155 \DeclareUTFSymbol\textboxbslash{"29C5}
9156 \DeclareUTFSymbol\textboxast{"29C6}
9157 \DeclareUTFSymbol\textboxcircle{"29C7}
9158 \DeclareUTFSymbol\textboxbox{"29C8}
9159 \DeclareUTFSymbol\textValve{"29D3}
9160 \DeclareUTFSymbol\textmultimapboth{"29DF}
9161 \DeclareUTFSymbol\textshuffle{"29E2}
9162 \DeclareUTFSymbol\textuplus{"2A04}
9163 \DeclareUTFSymbol\textbigdoublewedge{"2A07}
9164 \DeclareUTFSymbol\textbigdoublevee{"2A08}
9165 \DeclareUTFSymbol\textJoin{"2A1D}
9166 \DeclareUTFSymbol\textfatsemi{"2A1F}
9167 \DeclareUTFSymbol\textcircplus{"2A22}
9168 \DeclareUTFSymbol\textminusdot{"2A2A}
9169 \DeclareUTFSymbol\textdottimes{"2A30}
9170 \DeclareUTFSymbol\textdttimes{"2A32}
9171 \DeclareUTFSymbol\textodiv{"2A38}
9172 \DeclareUTFSymbol\textinvneg{"2A3C}
9173 \DeclareUTFSymbol\textsqdoublecap{"2A4E}
9174 \DeclareUTFSymbol\textcapdot{"2A40}
9175 \DeclareUTFSymbol\textsqdoublecup{"2A4F}
9176 \DeclareUTFSymbol\textdoublewedge{"2A55}
9177 \DeclareUTFSymbol\textdoublevee{"2A56}
9178 \DeclareUTFSymbol\textdoublebarwedge{"2A5E}
9179 \DeclareUTFSymbol\textveedoublebar{"2A63}
9180 \DeclareUTFSymbol\texteqdot{"2A66}
9181 \DeclareUTFCommand\textneqdot{\textlstrikethru\texteqdot}
9182 \DeclareUTFSymbol\textcoloncolonequals{"2A74}
9183 \DeclareUTFSymbol\textleqslant{"2A7D}
9184 \DeclareUTFCommand\textnleqslant{\textlstrikethru\textleqslant}
9185 \DeclareUTFSymbol\textgeqslant{"2A7E}
9186 \DeclareUTFCommand\textngeqslant{\textlstrikethru\textgeqslant}
9187 \DeclareUTFSymbol\textlessapprox{"2A85}
9188 \DeclareUTFCommand\textnlessapprox{\textlstrikethru\textlessapprox}
9189 \DeclareUTFSymbol\textgtrapprox{"2A86}
9190 \DeclareUTFCommand\textngtrapprox{\textlstrikethru\textgtrapprox}
9191 \DeclareUTFSymbol\textlneq{"2A87}
9192 \DeclareUTFSymbol\textgneq{"2A88}
9193 \DeclareUTFSymbol\textlnapprox{"2A89}
9194 \DeclareUTFSymbol\textgnapprox{"2A8A}
9195 \DeclareUTFSymbol\textlesseqqtr{"2A8B}
9196 \DeclareUTFSymbol\textgtreqqlss{"2A8C}
9197 \DeclareUTFSymbol\texteqslantless{"2A95}
9198 \DeclareUTFSymbol\texteqslantgtr{"2A96}
9199 \DeclareUTFSymbol\textleftslice{"2AA6}
9200 \DeclareUTFSymbol\textright slice{"2AA7}
9201 \DeclareUTFSymbol\textpreceq{"2AAF}
9202 \DeclareUTFCommand\textnpreceq{\textlstrikethru\textpreceq}
9203 \DeclareUTFSymbol\textsucceq{"2AB0}
9204 \DeclareUTFCommand\textnsucceq{\textlstrikethru\textsucceq}
9205 \DeclareUTFSymbol\textprecneq{"2AB1}
9206 \DeclareUTFSymbol\textsuccneq{"2AB2}
9207 \DeclareUTFSymbol\textpreceqq{"2AB3}
9208 \DeclareUTFCommand\textnpreceqq{\textlstrikethru\textpreceqq}
9209 \DeclareUTFSymbol\textsucceqq{"2AB4}
9210 \DeclareUTFCommand\textnsucceqq{\textlstrikethru\textsucceqq}
9211 \DeclareUTFSymbol\textprecneqq{"2AB5}
9212 \DeclareUTFSymbol\textsuccneqq{"2AB6}
9213 \DeclareUTFSymbol\textprecapprox{"2AB7}
9214 \DeclareUTFCommand\textnprecapprox{\textlstrikethru\textprecapprox}
9215 \DeclareUTFSymbol\textsuccapprox{"2AB8}
9216 \DeclareUTFCommand\textnsuccapprox{\textlstrikethru\textsuccapprox}
9217 \DeclareUTFSymbol\textprecnapprox{"2AB9}
9218 \DeclareUTFSymbol\textsuccnapprox{"2ABA}

```

```

9219 \DeclareUTFSymbol\textsubseteqq{"2AC5}
9220 \DeclareUTFCommand\textnsubseteqq{\textlstrikethru\textsubseteqq}
9221 \DeclareUTFSymbol\textsupseteqq{"2AC6}
9222 \DeclareUTFCommand\textnsupseteqq{\textlstrikethru\textsupseteqq}
9223 \DeclareUTFSymbol\textdashV{"2AE3}
9224 \DeclareUTFCommand\textndashV{\textlstrikethru\textdashV}
9225 \DeclareUTFSymbol\textDashv{"2AE4}
9226 \DeclareUTFCommand\textnDashv{\textlstrikethru\textDashv}
9227 \DeclareUTFSymbol\textDashV{"2AE5}
9228 \DeclareUTFCommand\textnDashV{\textlstrikethru\textDashV}
9229 \DeclareUTFSymbol\textdownmodels{"2AEA}
9230 \DeclareUTFCommand\textndownmodels{\textlstrikethru\textdownmodels}
9231 \DeclareUTFSymbol\textupmodels{"2AEB}
9232 \DeclareUTFCommand\textnupmodels{\textlstrikethru\textupmodels}
9233 \DeclareUTFSymbol\textupspoon{"2AEF}
9234 \DeclareUTFSymbol\textinterleave{"2AF4}
9235 \DeclareUTFSymbol\textsslash{"2AFD}
9236 \DeclareUTFSymbol\textpentagon{"2B20}
9237 \DeclareUTFSymbol\textvarhexagon{"2B21}
9238 \DeclareUTFSymbol\textjinferior{"2C7C}
9239 \DeclareUTFSymbol\textslashdiv{"2E13}
9240 \DeclareUTFSymbol\textinterrobangdown{"2E18}
9241 \DeclareUTFSymbol\textfivedots{"2E2D}
9242 \DeclareUTFSymbol\textPUheng{"A727}
9243 \DeclareUTFSymbol\textPUhookfour{"A72C}
9244 \DeclareUTFSymbol\textPUscf{"A730}
9245 \DeclareUTFSymbol\textPUaolig{"A735}
9246 \DeclareUTFSymbol\textoo{"A74F}
9247 \DeclareUTFSymbol\textcircumlow{"A788}
9248 \DeclareUTFSymbol\textfi{"FB01}
9249 \DeclareUTFSymbol\textfl{"FB02}
9250 \DeclareUTFSymbol\textGaPa{"1D13B}
9251 \DeclareUTFSymbol\textHaPa{"1D13C}
9252 \DeclareUTFSymbol\textViPa{"1D13D}
9253 \DeclareUTFSymbol\textAcPa{"1D13E}
9254 \DeclareUTFSymbol\textSePa{"1D13F}
9255 \DeclareUTFSymbol\textZwPa{"1D140}
9256 \DeclareUTFSymbol\textfullnote{"1D15D}
9257 \DeclareUTFSymbol\texthalfnote{"1D15E}
9258 \DeclareUTFSymbol\textVier{"1D15F}
9259 \DeclareUTFSymbol\textAcht{"1D160}
9260 \DeclareUTFSymbol\textSech{"1D161}
9261 \DeclareUTFSymbol\textZwdr{"1D162}
9262 \DeclareUTFSymbol\textMundus{"1F30D}
9263 \DeclareUTFSymbol\textMoon{"1F319}
9264 \DeclareUTFSymbol\textManFace{"1F468}
9265 \DeclareUTFSymbol\textWomanFace{"1F469}
9266 \DeclareUTFSymbol\textFax{"1F4E0}
9267 \DeclareUTFSymbol\textFire{"1F525}
9268 \DeclareUTFSymbol\textBicycle{"1F6B2}
9269 \DeclareUTFSymbol\textGentsroom{"1F6B9}
9270 \DeclareUTFSymbol\textLadiesroom{"1F6BA}
9271 \DeclareUTFCommand\textcopyleft{\textcircled\textrevc}
9272 \DeclareUTFCommand\textccsa{\textcircled\textcirclearrowleft}
9273 \DeclareUTFSymbol\textglqq{"201E}
9274 \DeclareUTFSymbol\textgrqq{"201C}
9275 \DeclareUTFSymbol\textglq{"201A}
9276 \DeclareUTFSymbol\textgrq{"2018}
9277 \DeclareUTFSymbol\textflqq{"00AB}
9278 \DeclareUTFSymbol\textfrqq{"00BB}
9279 \DeclareUTFSymbol\textflq{"2039}
9280 \DeclareUTFSymbol\textfrq{"203A}
9281 \DeclareUTFSymbol\textneg{"00AC}
9282 \DeclareUTFSymbol\textcdot{"00B7}

9283 </xunextra>

9284 <@=@=xeCJK>

```

5.22 **xeCJK.cfg**

9285 `<*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。

9286

9287 `</config>`

版本历史

v3.1.0	(2012/11/13 – 2012/11/21)	<code>\xeCJK@family</code> : 不将参数完全展开。	113
General: 放弃对 <code>\outer</code> 宏的特殊处理。	1	<code>_xeCJK_check_single_space:NN</code> : 使用 <code>\xeCJK_if_CJK_class:NTF</code> 来代替 <code>\int_case:nnn</code> 判断是否是 CJK 字符类。	65
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。	103	<code>_xeCJK_family_unknown_warning:n</code> : 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。	95
改用 <code>indentfirst</code> 宏包处理缩进的问题。	110	v3.2.0	(2013/04/14 – 2013/05/22)
取消 <code>\protect</code> 的外部宏限制。	126	General: 增加 IVS 字符类用于处理异体字选择符。	27
删除多余的 <code>default-itcorr</code> 结点。	45	增加 Verb 选项。	103
使用 <code>xtemplate</code> 宏包的机制来组织标点符号的处理。	72	<code>\setCJKmonofont</code> : 定义中加入 <code>\normalfont</code> 。	96
LocalConfig: 增加 LocalConfig 选项用于载入本地配置文件。	108	<code>_xeCJK_Boundary_and_FullLeft_glue:N</code> : 当全角左标点前面是 <code>hlist</code> 、 <code>none</code> 、 <code>glue</code> 和 <code>penalty</code> 等节点时, 压缩其左空白。	56
<code>\xeCJK@fix@penalty</code> : 采用通过不修改原语 <code>\</code> 的方式对修复倾斜校正。	113	<code>\l_xeCJK_family_tl</code> : 不将其初始化为 <code>\CJKfamilydefault</code> 。	95
<code>_xeCJK_fallback_loop:nnNN</code> : 调整备用字体的循环方式。	85	<code>\xeCJK_FullLeft_and_Default::</code> 修正 <code>xeCJK</code> 使西文在部分情况下无法断词的问题。	55
<code>\xeCJK_glyph_if_exist:N</code> : 改进 <code>fontspec</code> 宏包中定义的 <code>\font_glyph_if_exist:NnTF</code> 。	23	<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑到 <code>\spacefactor</code> 和 <code>\xspaceskip</code> 的情况。	23
<code>\xeCJK_hook_for_ulem::</code> 简化对 <code>ulem</code> 宏包的兼容补丁。	127	v3.2.1	(2013/05/29)
<code>\c_xeCJK_space_skip_tl</code> : 字间空格考虑 <code>\spaceskip</code> 不为零的情况。	23	General: 调整 Verb 选项: 在命令 <code>\verb</code> 里使用时, 不破坏标点禁则, 增加值 <code>env+</code> 。	103
<code>_xeCJK_switch_font:nn</code> : 改进定义, 加快切换速度。	92	v3.2.2	(2013/05/30 – 2013/06/04)
<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKVerbAddon</code> 用于抄录环境中的间距调整。	104	General: 修正某些重音不能正确显示的问题。	1
v3.1.1	(2012/12/02 – 2012/12/13)	增加小宏包 <code>xeCJK-listings</code> , 用于支持 <code>listings</code> 宏包。	145
General: 不再依赖 <code>xpatch</code> 宏包。	1	<code>_xeCJK_ulem_CJK_and_FullRight_glue:N</code> : 修正下划线不能跳过全角右标点的问题。	135
对于与 <code>xltextra</code> 的冲突给出错误警告。	113	v3.2.3	(2013/06/04 – 2013/06/11)
增加 <code>NewLineCS</code> 和 <code>EnvCS</code> 选项。	66	General: 不再改变 CJK 字符类的 <code>\catcode</code> 。	35
增加小宏包 <code>xeCJKintef</code> , 用于处理下划线的问题。	127	根据 $\text{\XeTeX}$ 的脚本重新整理全角标点符号。	29
<code>CheckFullRight</code> : 处理全角右标点之后的断行问题。	61	解决 <code>CheckSingle</code> 选项与 <code>tablists</code> 宏包的冲突。	65
<code>InlineEnv</code> : 改变行内环境的设置方式, 从而使用 <code>\str_case_x:nnn</code> 代替原来的 <code>\clist_if_in:NnTF</code> 来判断是否是行内环境。	66	提供四个 <code>TECkit</code> 映射文件用于句号转换和简繁互换。	1
<code>PlainEquation</code> : 增加 <code>PlainEquation</code> 选项。	65	完善对 <code>listings</code> 宏包的支持。	145
<code>\xeCJK@family</code> : 修改主要 CJK 字体族的自动更新方式。	113	<code>_xeCJK_listings_initial_hook::</code> 解决 <code>listings</code> 环境中代码行号输出不正确的问题, 并解决在其中跨页时对页眉和页脚的影响。	145
<code>_xeCJK_check_single_aux:nnNw</code> : 改进定义, 减少使用 <code>peek</code> 函数的次数。	64	<code>_xeCJK_listings_process_Default:nN</code> : 在 <code>listings</code> 环境中对 <code>\charcode</code> 大于 255 的字符根据其 <code>\catcode</code> 区分 <code>letter</code> 和 <code>other</code> 。	147
<code>_xeCJK_check_single_space:NN</code> : <code>CheckSingle</code> 支持段末“汉字 + 汉字 + 空格 + 汉字/标点”的形式。	65	<code>_xeCJK_restore_shipout_CJKsymbol::</code> 解决 <code>\CJKunderdot</code> 跨页使用时影响到页眉页脚的问题。	144
<code>\xeCJK_hook_for_ulem::</code> 完全处理下划线里的标点符号的有关问题。	127	<code>_xeCJK_ulem_FullLeft_and_CJK::</code> 修正全角左标点后下划线与 <code>\CJKunderdot</code> 连用时结果不正常的问题。	136
<code>\xeCJK_peek_catcode_ignore_spaces:NTF</code> : 新增有省略空格标识的 <code>peek</code> 函数。	25	<code>\xeCJKVerbAddon</code> : 新增 <code>\xeCJKOffVerbAddon</code> 用于局部取消 <code>\xeCJKOffVerbAddon</code> 的影响; 并解决跨页使用时影响到页眉页脚的问题。	104
<code>\xeCJK_save_class:nn</code> : 使用 <code>\xeCJK_save_class:nn</code> 保存 $\text{\XeTeX}$ 预定义的字符类别。	27	v3.2.4	(2013/06/23 – 2013/07/06)
<code>\xeCJK_set_char_class:nnn</code> : 在文档中设置字符类别时不重复设置 <code>\catcode</code> 。	35	General: 不再使用 <code>CJKnumber</code> 选项, 可以在 <code>xeCJK</code> 之后直接使用 <code>CJKnumb</code> 宏包得到中文数字。	126
<code>_xeCJK_set_char_class_eq:nn</code> : 交换参数的顺序。	35	改进获取分区字体属性的办法。	87
<code>_xeCJK_set_verb_exspace::</code> 调整间距的计算方法。	106	解决使用 <code>CheckSingle</code> 时, 某些 <code>\CJKglue</code> 不能被正确加入的问题。	65
<code>\xeCJKnobreak</code> : 增加 <code>\nobreak</code> 的 <code>xeCJK</code> 版本。	63	尽量移除用作判断标志的 <code>\kern</code> 。	45
v3.1.2	(2012/12/27 – 2013/01/01)	内部调整分区字体的设置方法。	87
General: 解决在下划线状态下使用 <code>\makebox</code> 时的错误。	132		
修正非 <code>\UTFencname</code> 编码下面 <code>xunicode</code> 重定义的 <code>\nobreakspace</code> 会失效的问题。	113		
修正重定义 <code>\CJKfamilydefault</code> 无效的问题, 恢复容错能力。	99		

使 listings 的 breaklines 选项对 CJK 字符类可用, 并保持标点符号的禁则。	148	修正 unicode-letters.tex 中谚文符号 \catcode 不准的问题。	35
使用 AllowBreakBetweenPuncts 时, 相应标点符号仍能与边界对齐。	54	\Url@MathSetup: 使通过 \UrlFont 等命令设置的 CJK 字体生效。	111
修正 xeCJKfntef 与 natbib 等的冲突。	127	_xeCJK_check_single_aux:n:Nw: 与 \CJKspace 兼容。	64
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。	53
\addCJKfontfeatures: 可以单独增加当前各个分区字体的属性。	97	\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。	100
CJKfilltwosides: 改用 minipage 和 L ^A T _E X 表格 (tabular) 来实现。	144	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。	56	General: 启用 xunicode 中的带圈数字和字母设置。	163
_xeCJK_fallback_loop:nnNN: 使 \CJKfamilydefault 的 FallBack 设置全局可用。	85	\DeclareUTFmathsymbols: 修正 \UseMathAsText 的功能, 恢复 \hbar 和增加以 text 打头的文本符号命令。	153
_xeCJK_set_verb_exspace:: 当计算得出的间距为负时, 缩小 CJK 字体。	106	_xeCJK_nobreak_skip:: 禁止在 \verb 中断行。	103
\xeCJK_tl_remove_outer_braces:n: 去掉外层分组括号时, 移除空格, 避免死循环。	22	\xeCJKVerbAddon: 增加是否是等宽字体的判断。	104
\xeCJK_token_value_charcode:N: 考虑 charcode 超出 BMP 的情况。	25	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。	151
General: 恢复 \nobreakspace 的原始定义。	113	增加 xunicode-extra.def 中, 用于加入 puenc.def 中的符号定义。	167
解决 fixltx2e 和 amsthm 的冲突。	113	\DeclareEncodedCompositeAccents: 修正 xunicode 中的错误定义。	157
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。	41	\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。	117
增加小宏包 xunicode-addon, 为 xunicode 提供判断字符是否存在的功能。	151	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。	107	\CJKaddEncHook: 使用 CJKnumb 时, 让 \Unicode 有定义。	126
Verb: 微调定义。	103	\DeclareUTFDoubleEncodedAccent: 改进 \t 等的定义方式。	157
_xeCJK_Boundary_and_FullLeft_glue:N: 细化全角左标点是否位于段首的判断。	56	\DeclareUTFDoubleEncodedSymbol: 改进 \sliding 等的定义方式。	157
增加对 enumitem 宏包修改的 \item 的判断。	56	\DeclareUTFTIPACCommand: 检查 \t 和 \sliding 的参数是否以 \textipa 开头。	162
_xeCJK_math_robust:N: 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。	112	LoadFandol: 当没有设置字体时, 使用 Fandol 字体系列。	99
\xeCJKVerbAddon: 禁止自动换行, 与西文一致。	104	v3.2.11 (2014/03/14 – 2014/04/10)	
v3.2.6 (2013/07/29 – 2013/08/15)		General: 删除 \xeCJKcaption。	126
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。	88	左右角括号 U+2329 和 U+232A 是西文标点符号。	29
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。	95
为 \mathrm 减少一个可能的数学字体族。	112	indentfirst: 放弃 indentfirst 和 CJKnumber 选项。	108
\AtEndUTFCommand: 可以指定特定符号命令使用的钩子。	161	\xeCJK_add_to_shipout:n: 不再使用内部名字。	22
\xeCJK_CJK_and_Boundary:w: 更好的处理边界是 \relax 的情况。	48	v3.2.12 (2014/05/12)	
_xeCJK_math_robust:N: 考虑 ulem 对 \MakeRobust 的不当定义。	112	General: 更新 \int_to_Hex:n。	85
考虑 \math 和 \ensuremath。	112	新增 RubberPunctSkip 选项。	69
\xeCJK_set_mathfont:: 设置粗体时先检查对应字体是否存在。	100	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。	99
General: 标点宽度设置禁用比例选项的值改为 nan。	79	\xeCJK_set_mathfont:: 修复参数类型错误。	100
处理 AllowBreakBetweenPuncts 与 xeCJKfntef 的兼容问题。	54	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。	68	General: xeCJKfntef 不再依赖 CJKfntef。	127
使用 everypage 往 \shipout 盒子里加钩子。	22	解决下划线前后没有 \CJKglue 或 \CJKecglue 的问题。	127
		完善 \varCJKunderline 的实现。	127
		v3.2.15 (2014/11/07 – 2014/11/10)	
		General: xeCJKfntef 增加 hidden 选项。	127
		把 REVERSE SOLIDUS(U+005C)、HYPHEN-MINUS (U+002D)和 EN DASH(U+2013)归入 NormalSpace 类。	29
		增加 HangulJamo 字符类。	28

\CJKunderanyline: 完善选项。	140	v3.4.1	(2016/05/21 – 2016/08/18)
_xeCJK_listings_initial_hook:: 修正 breaklines 无效的问题。	145	General: 补充 Unicode 9.0 的西夏文。	31
\xeCJKfntefon: 完善选项。	138	新的下划线选项 textformat。	127
v3.2.16	(2014/11/20 – 2014/12/16)	修复 CJKspace 功能失效。	48
General: 不再依赖 everypage 宏包。	22	v3.4.2	(2016/10/19)
修复 \hbar。	111	General: 避免在破折号之间折行。	38
整理 xCJKecglue 的部分代码。	47	\xeCJK_clear_Boundary_and_CJK_toks:: 提高效率, 避免重复循环。	37
v3.3.0	(2014/12/26)	v3.4.3	(2016/10/27 – 2016/11/18)
General: 不把 NS 类中的一些有禁则的日文归入 FullRight 类。	29	\CJKfontspec: 允许字体属性可选项在后的新语法。	97
不把小写日文假名归入 FullRight 类。	31	\setCJKfallbackfamilyfont: 允许字体属性可选项在后的新语法。	86
\c_xeCJK_PR_chars_clist: 不把 U+20A9 归入 CJK 的 PR 类。	29	\setCJKmathfont: 允许字体属性可选项在后的新语法。	100
v3.3.1	(2015/01/22 – 2015/05/08)	\setCJKmonofont: 允许字体属性可选项在后的新语法。	96
General: IVS 字符类更名为 CM。	27	_xeCJK_long_punct_kerning:N: 考虑破折号边界为负值的情况。	74
删去 fixltx2e 和 amsthm 的冲突补丁。	113	v3.4.4	(2016/11/30)
新选项 WidowPenalty。	64	General: 不压缩长标点与其他标点的间距。	79
\CJKaddEncHook: 应用 0.99992 版的新原语 \Ucharcat。	126	v3.4.5	(2017/01/02)
LoadFandol: 为方便 MacTeX 用户, Fandol 字体改用文件名。	99	General: 更新 L ^A T _E X3 的过时用法。	112
\xeCJK_check_single_cs:Nn: 补充可能遗漏的空格。	65	v3.4.6	(2017/02/23)
\c_xeCJK_CM_chars_clist: 补充音调符号。	33	\xeCJK@family: 将族名参数完全展开, 以解决与 fontspec 2017/01/24 v2.5d 的兼容问题。	113
_xeCJK_listings_initial_hook:: 解决 prebreak 和 postbreak 功能失效的问题。	145	v3.4.7	(2017/03/20)
_xeCJK_listings_process_Default:nN: 对 listings 的字符扩展不影响到其符号表中的七位或八位字符。	147	General: 简化 CheckSingle 的实现, 不再展开宏。	64
_xeCJK_math_robust:N: 兼容 L ^A T _E X 2 _ε 2015。	112	v3.4.8	(2017/05/15)
\xeCJK_token_value_charcode:N: 0.99992 版修复了 \meaning 的 Bug。	25	General: 转义 \lstineline 参数中的 \ ₁₂ 。	151
\g_xeCJK_xetex_allocator_int: 兼容 L ^A T _E X 2 _ε 2015。	114	v3.5.0	(2017/07/19 – 2017/07/22)
v3.3.2	(2015/05/15)	General: 补充 Ext-F。	31
General: 随 Unicode 7.0 更新简繁体映射。	1	常数 \c_minus_one 已过时。	1
\g_xeCJK_xetex_allocator_int: \xe@alloc@intercharclass 总是有定义的。	114	使用 lazy 函数对 Boolean 表达式进行最小化运算 (L ^A T _E X3 2017/07/19)。	1
v3.3.3	(2015/05/30 – 2016/02/01)	v3.5.1	(2017/11/16)
General: 把 EN DASH(U+2013)作为半字线连接号归入 FullRight 类。	29	General: 修正 fallback 字体后无法忽略空格的错误。	85
补充 Ext-E。	31	v3.6.0	(2018/01/13 – 2018/01/24)
不再把 U+2015 和 U+2500 归入 FullRight 类。	29	General: Default 类与 MiddlePunct 之间不应该有 \CJKglue。	60
更新 L ^A T _E X3 代码。	1	把 TWO-EM DASH (U+2E3A) 归入 FullRight 类和设为 LongPunct 与 MiddlePunct。	29
兼容 L ^A T _E X 2 _ε 2016/02/01 的字符类设置。	28	将全角浪线 U+FF5E 等连接号归入 FullRight 类和设为 MiddlePunct。	29
解决与 microtype 宏包的兼容问题。	126	解决标点中间被隔开的禁则与压缩问题。	58
使用新的 Unicode 编码名称 TU。	152	同步 L ^A T _E X3 2017/12/16。	1
CJKfilltwosides: 确保进入水平模式。	144	新增 PunctFamily 选项支持对汉字标点单独切换字体。	94
v3.3.4	(2016/02/07)	修正标点同为 LongPunct 与 MiddlePunct 时的实现错误。	60
General: 兼容 X _Y L ^A T _E X 0.99994 的边界字符类。	28	总允许长标点与其他标点之间折行。	53
v3.4.0	(2016/05/01 – 2016/05/13)	v3.6.1	(2018/02/25 – 2018/02/27)
General: RubberPunctSkip 选项有新的值 plus 和 minus。	69	General: 减少 bool 运算。	1
CJKmath 功能也支持分区字体。	101	\xeCJK_if_last_punct:TF: 细化判断。	58
标点符号的压缩量能伸长到原始空白, 能收缩到较小边距。	80	v3.7.0	(2018/03/12 – 2018/03/18)
改进 xCJKecglue 的实现。	40	General: 补充定义 \texthyphenationpoint 和 \texttwoemdash。	163
\xeCJK_set_mathfont:: CJKmath 的字符范围遵从 \xeCJKDeclareCharClass 的设置。	100	不再默认引入 xunicode 宏包。	109
		对 \nobreakspace 的恢复放到 xunicode-addon 中处理。	113

修正长标点被隔开时的压缩处理错误。.....	74	将 CJKfntef 包替换为 xeCJKfntef 包。.....	21
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。...	161	\declare@file@substitution。.....	21
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。.....	31
CJKfilltwosides* 环境, 不再依赖 environ 包。.....	144	同步 Unicode 14.0。.....	31
简化 CJKspace 的实现, 并修复错误。.....	48	v3.8.9 (2022/05/26)	
解决与 microtype 宏包的兼容问题。.....	163	General: 修正居中标点悬挂错误。.....	78
删除定义新字体族时过滤重复选项的功能。.....	89	_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。.....	56
\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。.....	55	General: 不直接依赖 xparse 和 l3keys2e。.....	21
_xeCJK_patch_tuenc_composite:: 修复补丁错误。..	116	修复西文的 character protrusion 功能。.....	46
v3.7.3 (2019/04/15)		v3.9.1 (2022/07/28 – 2022/08/02)	
General: 补充日文假名扩展。.....	31	General: 简化部分内部实现。.....	45
修复 penalty 数值错误。.....	58	修复下划线中数学公式的错误处理。.....	132
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。.....	68	General: NoBreakLongPunct 在右侧时禁止折行。.....	53
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。.....	126	_xeCJK_check_for_glue_skip: 中处理 \colorbox	
删除 realscripts 补丁。.....	113	等命令右侧间距(#831)。.....	120
删除 \hbar 补丁。.....	111	补丁 l3color 后端的 _color_select:N 和	
删除 \mathrm 补丁。.....	112	_color_backend_reset:, 使 l3color 接口的颜色切换	
应用 \fp_if_nan:nTF。.....	79	也能正确保持 xeCJK 间距(#832)。.....	121
应用 \peek_remove_spaces:n。.....	25	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。.....	107	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)。.....	120
General: 修复 \l_xeCJK_current_font_tl 标记错误。..	92	撤回对 \char 的重定义, 改为提供 \xeCJKchar 新命令,	
应用 \shapedefault。.....	101	避免破坏依赖	
v3.8.2 (2020/02/17)		let	
General: 避免导言区字体警告。.....	96	保存 \tn {char} 原语的宏包(\#800)。hdclindex6127117	
修复分区字体错误。.....	92	改正拼写错误 _xeCJK_glue_node:n。.....	45
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。.....	131	消除不必要的标点压缩间距。.....	29
补充 U+02EA 和 U+02EB。.....	31	使用 \l_keys_key_str 和 \l_keys_choice_str 替代	
兼容 unicode-math 和 CJKmath 选项。.....	118	已废弃的 _tl 版本(#806)。.....	120
取消 xeCJKfntef 的初始彩色设置。.....	142	提升 L ^A T _E X3 最低版本要求至 2025/10/09。.....	20
删除 _nopar。.....	1	提升版本号至 v3.10.0。.....	1
同步 Unicode 13.0。.....	31	通过在 \Hy@BeginAnnot 中保存并选择性恢复 xeCJK	
修复 xCJKecglue 选项。.....	58	节点标记, 同时清除旧标记, 解决目录中链接注释起始处	
依赖 ctexthook 宏包。.....	21	的虚假 ecglue(#810), 并为 \ref 提供前侧 ecglue	
重构 PunctStyle 选项, 完全展开参数。.....	82	(#809)。.....	119
v3.8.4 (2020/05/31 – 2020/06/04)		同步 Unicode 15.0。.....	31
General: 重构后备字体的实现, 修正标点符号无后备字		同步 Unicode 15.1。.....	31
体的问题。.....	84	同步 Unicode 17.0。.....	31
\xeCJK_font_gset_to_current:N: 不缓存 \nullfont。..	23	为 color/xcolor 添加兼容补丁, 在颜色切换 whatsit 后重	
v3.8.5 (2020/06/25 – 2020/06/26)		放 xeCJK 节点标记, 修复 \textcolor 后 CJKecglue 丢	
General: 进一步兼容 microtype。.....	119	失的问题(#315, #803)。.....	120
_xeCJK_bound_type_1_glue:Nn: 增加盒子高度判断。..	56	为 hypdoc 的 \HD@target 添加补丁, 在它产生的 hbox	
v3.8.6 (2020/10/17 – 2020/10/18)		之后重放 xeCJK 节点标记, 修复 l3doc 中 \cs、\meta 等	
General: 兼容 L ^A T _E X 2020/10/01 的 NFSS 钩子机制。...	113	命令后 CJKecglue 丢失或保留为原始空格的问题	
正确还原标点符号后的 penalty 状态。.....	58	(#873)。.....	122
v3.8.7 (2021/06/04 – 2021/06/09)		为 mtpro2 提供兼容补丁, 使大花括号内部的 \char 不	
General: 更好地兼容 CJKnumb。.....	126	被 interchar 拦截(#407)。.....	118

为 url 的 <code>\Url@FormatString</code> 添加补丁, 在进入数学模式前 <code>drain</code> 缓存的 <code>CJKecglue</code> , 修复 CJK 文字与 <code>\url</code> 命令之间间距丢失的问题(#880)。	122	新增 <code>glue</code> 分支处理, 修复 <code>xeCJKfntef</code> 命令右侧空格问题。	42
文档字体从 Source Han Serif 统一为 Noto Serif CJK (#686)。	1	<code>\xeCJK_CJK_and_Boundary:w</code> : 当显式触发 <code>CJK→Boundary</code> 转换时, 设置	
新增 <code>NoBreakLongPunct</code> 属性, 禁止在省略号等长标点前断行。	60	<code>\g__xeCJK_glue_check_pending_bool</code> 以启用后续的 <code>glue</code> 检查, 修复前 {中} 后类型排版中多余的	
新增 <code>\xeCJKchar</code> , 绕过 <code>interchar</code> 机制(#407)。	117	<code>\CJKecglue</code> (#831)。	48
新增实验性 <code>experiment/halfright-prebreakpenalty</code> 选项(#811)。	50	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 <code>glue</code> , 改为只留标记由后续恢复路径	
修复 <code>xCJKecglue</code> 错误。	45	决定间距(#324)。	48
修复 <code>\set@color</code> 补丁在无真实节点标记时未清除全局状态, 导致首次 <code>\textcolor</code> 在标点或段首后插入虚假 <code>ecglue</code> 的问题(#807)。	120	<code>\xeCJK_fntef_sbox:n</code> : 保存并恢复	
修复 <code>\textcolor</code> 包裹 <code>ulem</code> 类下划线命令时 CJK 字间距异常的问题(#830)。	1	<code>\g__xeCJK_last_node_tl</code> , 避免 <code>hbox</code> 内渲染装饰符号时 <code>interchar toks</code> 污染全局节点标记, 修复 <code>xeCJKfntef</code>	
修复 <code>\textcolor</code> 等 <code>whatsit</code> 节点导致 <code>CJKecglue</code> 丢失的问题。	45	与 <code>\textcolor</code> 组合使用时多余的 <code>CJKecglue</code> 。	141
修复 <code>\textnleqslant</code> 的定义, 修复 <code>\textnbacksim</code> 和 <code>\textnlessapprox</code> 定义里的自引。	167	<code>\xeCJK_make_node:n</code> : 新增	
修复 <code>\textsbleftarrow</code> 的定义。	163	<code>\g__xeCJK_glue_check_pending_bool</code> 用于标记	
修复 <code>\xeCJKnobreak</code> 错误。	45	<code>xeCJKfntef</code> 组结束后的首次 <code>glue</code> 探测。	46
修复拼错的命令名, <code>\cyrreref</code> → <code>\cyrrerev</code> 和		新增 <code>\g__xeCJK_reset_color_pending_bool</code> 用于标记	
<code>\textDiamandSolid</code> → <code>\textDiamondSolid</code> 。	167	<code>\reset@color</code> 在 <code>hlist</code> 上下文中触发, 延迟到	
修复颜色补丁在 <code>\g__xeCJK_last_node_tl</code> 被无关的		<code>__xeCJK_check_for_glue_skip</code> : 中处理 <code>\colorbox</code>	
<code>\set@color</code> 清空后仍尝试重放节点导致 <code>Missing number</code> 错误的问题, 影响 <code>listings</code> 与 <code>\rulecolor</code> 组合	120	等命令右侧的 <code>color pop</code> <code>whatsit</code> (#831)。	46
修复在 <code>\lstinline</code> 参数中使用非		<code>\c__xeCJK_middle_dot_prop</code> : 在 <code>\Pifont</code> 中先进入水平	
# 的 <code>catcode~6</code> 字符(如 <code>\textttt {\textbackslash catcode~6}</code>)		模式, 防止 <code>\makexeCJKinactive</code> 在垂直模式下通过分	
时, 输出错字的问题(#879)。	51	页泄漏到输出例程(#688)。	117
<code>hd-clin-dex7878145</code>		<code>__xeCJK_punct_boundary_guard::</code> 新增	
修复字体切换组(如 <code>\textttt</code> 、 <code>\zihao</code>)导致 <code>CJKecglue</code>		<code>experiment/punct-measure-fix</code> 选项: 段落模式下通	
使用错误字体度量的问题。	45	过 <code>para/end</code> 钩子恢复被 <code>\unskip</code> 移除的标点补偿	
修复组合符号 <code>U+04AA</code> 的定义。	167	<code>glue</code> (#859)。	51
在下划线内 CJK 分组切换时保存/恢复字体状态		修复 <code>tabular</code> 中 <code>\unskip</code> 吞掉标点补偿 <code>glue</code> 的问题	
(#465)。	133	修复 <code>tabular</code> 中 <code>\unskip</code> 吞掉标点补偿 <code>glue</code> 的问题	
<code>\CJKfontspec: Breaking: \newCJKfontfamily</code> 定义的字		<code>\xeCJK_under_symbol:nnnnnn</code> : 在独立模式的末尾设置	
体切换命令改为局部定义, 与 <code>\newcommand</code> 行为一致。		<code>\g__xeCJK_glue_check_pending_bool</code> , 修复	
在分组内调用时, 命令不再泄漏到分组外(#751)。	97	<code>\CJKunderdot</code> 等命令右侧空格问题(#826)。	143
<code>\CJKunderanysymbol</code> : 移除 <code>\ignorespaces</code> , 修复命令后		<code>\xeCJKsetcharclass: Deprecated:</code>	
空格丢失(#465)。	142	<code>\xeCJKsetcharclass</code> 已弃用, 调用时报错并提示改用	
<code>\CJKunderdot</code> : 移除 <code>\ignorespaces</code> , 修复命令后空格丢		<code>\xeCJKDeclareCharClass</code> (#709)。	111
失(#465)。	143		
<code>\xeCJK@family: Breaking: 移除对 $\TeX$ 2_ε 2020/10/01</code>		v3.10.1	(2026/06/29 – 2026/07/01)
之前版本的字体钩子兼容代码(#746)。	113	General: 提升版本号至 v3.10.1。	1
<code>\xeCJK_Boundary_and_Default::</code> 移除通用 <code>whatsit</code> 恢		为 <code>l3doc</code> 的 <code>__codedoc_meta:n</code> 添加补丁, 将参数内容	
复, 避免 <code>hyperref</code> 链接注释等 <code>whatsit</code> 节点导致		包入 <code>\hbox:n</code> , 修复 <code>\meta</code> 内首字符前出现多余	
<code>CJKecglue</code> 在错误位置插入(#803)。配合 <code>\set@color</code>		<code>CJKecglue</code> 的问题(#920)。	124
定点补丁保持 <code>\textcolor</code> 场景的正确性(#315)。	40	为 $\TeX$ 核心命令 <code>\verb</code> 添加补丁, 在进入 <code>verb</code> 内容前	
<code>\xeCJK_check_for_glue::</code> 当 <code>glue</code> 下方是 <code>hlist</code> (<code>\mbox</code> 等		<code>drain</code> 缓存的 <code>CJKecglue</code> , 修复 CJK 文字与 <code>\verb</code> (或	
命令产生的 <code>hbox</code>)时, 不依赖 <code>boolean</code> , 直接通过		<code>shortvrb</code> 的短记号)之间间距丢失的问题(#910)。	123
<code>\g__xeCJK_last_node_tl</code> 判断前方内容类型, 修复		为 <code>biblatex</code> 的 <code>\blx@pagetracker</code> 添加补丁, 清空	
<code>\mbox</code> 、 <code>\colorbox</code> 等 <code>hbox</code> 命令右侧多余 <code>inter-word</code>		<code>\g__xeCJK_last_node_tl</code> , 修复中文参考文献条目首字	
<code>glue</code> (#831)。	42	符前出现多余 <code>CJKecglue</code> 的问题(#931)。	125
		v3.10.2	(2026/07/03 – 2026/07/06)
		General: <code>PoZheHao</code> 类字符同样按全角右标点处理(#382)。	71
		<code>\verb</code> 组关闭时用 <code>\setlanguage</code> 主动排出 <code>language</code>	
		<code>whatsit</code> , 修复 <code>\verb</code> 右侧源码空格场景下 <code>CJKecglue</code>	
		双倍的问题(#919)。	123
		提升版本号至 v3.10.2。	1
		为 <code>doc</code> 的 <code>\meta</code> 添加补丁, 将参数内容包入 <code>\hbox:n</code> ,	
		修复 <code>\meta</code> 内首字符前出现多余 <code>CJKecglue</code> 的问题	124

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	shaping, 在相邻音节间恢复 \CJKglue(#158)。	39
文档字体启用 Language 与 PoZheHaoLigature, 破折号按全角字形合字输出(#382)。	1	开明式句末点号的宽度由 0.8em 改为全角(#975)。	83
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)。	69	全角格式启用相邻标点边界优化, 并保留左标点后续右标点的自然空白(#975)。	83
新增 PoZheHaoLigature 选项, 支持字体的破折号合字功能(#382)。	69	提升版本号至 v3.10.3。	1
增加 PoZheHao 字符类, 支持字体的破折号合字功能(#382)。	28	新增 CJLineBreak 选项, 为日文小假名提供可选严格行首禁则(#165)。	69
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)。	53	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)。	39
_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)。	74	新增 enabled-left-right-kerning 标点格式选项(#975)。	13
_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)。	76	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)。	119
\xeCJK_punct_margin_process:NN: 未启用合字的破折号两端各补偿一完整份空白, 保证连用的破折号占两个汉字宽(#382)。	76	\CJKsout: 使用 \cleaders 使删除线相对正文居中(#967)。	139
\xeCJKResetPunctClass: 重置后恢复 LatinPunct 的设置(#389)。	35	\CJKunderanyline: 使用 \cleaders 使自定义线相对正文居中(#967)。	140
重置后恢复 PoZheHaoLigature 的设置(#382)。	35	\CJKunderdblline: 使用 \cleaders 使双下划线相对正文居中(#967)。	139
v3.10.3 (2026/07/12 – 2026/07/13)		\CJKunderline: 使用 \cleaders 使下划线相对正文居中, 避免起始位置改变时产生水平偏移(#531)。	138
General: 将朝鲜文字母分为 L/V/T 三类, 在音节内保持		\CJKunderwave: 使用 \xleaders 使波浪线相对正文对齐, 同时保持字间连接连续(#967)。	138
		\CJKxout: 使用 \cleaders 使斜删除线相对正文居中(#967)。	139

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	
\\	5, 6, 7, 13, 14, 15, 16, 30, 365, 366
\<font-switch>	8
A	
\addCJKfontfeatures	8
\AddToHook	1488, 1490
AllowBreakBetweenPuncts	6
\AtBeginDocument	69
\AtEndOfPackage	432, 728, 741, 752, 776, 806
AutoFakeBold	4, 7
AutoFakeSlant	4, 7
AutoFallBack	4
B	
bool commands:	
\bool_gset_false:N	974, 991, 997, 1013, 1486, 1489, 1495
\bool_gset_true:N	1333, 1473
\bool_if:NTF	101, 319, 609, 611, 632, 972, 990, 995, 1012, 1060, 1315, 1321, 1360, 1364, 1418, 1425, 1471, 1492
\bool_lazy_or:nnTF	556
\bool_new:N	52, 110, 269, 614, 615, 636, 726, 1225, 1226, 1289, 1309, 1415, 1464
\bool_set_false:N	244, 1263, 1301
\bool_set_true:N	103, 251, 722, 1252, 1274, 1295
Boundary	368
box commands:	
\box_new:N	50
\box_wd:N	223
boxdepth	16
C	
char commands:	
\char_generate:nn	435
\char_set_catcode_ignore:n	331, 332, 333, 334, 335
\char_set_catcode_letter:n	574
CheckFullRight	6
CheckSingle	3
CJK	368
\CJKecglue	835, 893, 1317, 1353, 1361
CJKecglue	3, 1242
\CJKfamily	8
\CJKfamilydefault	9
CJKfilltwosides	17
\CJKfontspec	8

`\CJKglue` 804, 978, 982, 1019,
 1027, 1041, 1045, 1063, 1106, 1201, 1207, 1237, 1303, 1389
`CJKglue` 3, 1233
`CJKmath` 3
`\CJKrmdefault` 8
`\CJKsfdefault` 8
`\CJKsout` 15
`CJKspace` 3, 1290
`\CJKsymbol` 795, 826, 933, 1391, 1397, 1403
`\CJKttdefault` 8
`\CJKunderanyline` 16
`\CJKunderanysymbol` 17
`\CJKunderdblline` 15
`\CJKunderdot` 15
`\CJKunderline` 15
`\CJKunderwave` 15
`\CJKxout` 15
`CJLineBreak` 5
`CJStarter` 387
clist commands:
`\clist_const:Nn` 399, 401, 403, 404, 411, 413, 418, 426,
 428, 430, 440, 442, 443, 452, 453, 462, 512, 519, 521, 523
`\clist_gclear:N`
 601, 602, 603, 604, 618, 619, 620, 621, 622, 623, 624, 625
`\clist_gconcat:NNN` 539, 553
`\clist_map_function:NN` 437
`\clist_map_function:nN` 713
`\clist_map_inline:Nn` 541, 591
`\clist_map_inline:nn`
 33, 38, 754, 765, 767, 817, 830, 877, 1407, 1478
`\clist_new:N` 54, 345, 357
`\clist_set:Nn` 534
CM 387
cs commands:
`\cs_end:` 307
`\cs_generate_variant:Nn` 553, 643, 656, 662
`\cs_gset_eq:NN` 130, 141
`\cs_gset_protected:Npe` 691
`\cs_if_eq:NNTF` 141
`\cs_if_exist:NTF` 44, 94
`\cs_if_exist_use:NTF` 1134
`\cs_new:Npn` 111, 116,
 200, 270, 272, 279, 285, 296, 310, 361, 525, 644, 1161, 1163
`\cs_new_eq:NN`
 266, 267, 268, 362, 727, 857, 910, 1065, 1227, 1287
`\cs_new_protected:Npe` 1228
`\cs_new_protected:Npn` 55,
 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 72, 74, 76, 78, 80,
 82, 84, 91, 107, 127, 129, 131, 138, 140, 209, 212, 228,
 233, 237, 248, 318, 338, 351, 532, 537, 554, 578, 588,
 638, 649, 651, 657, 663, 676, 687, 689, 705, 719, 798,
 837, 839, 845, 858, 897, 899, 905, 935, 953, 970, 988,
 1005, 1010, 1016, 1036, 1052, 1058, 1066, 1081, 1105,
 1139, 1146, 1156, 1165, 1170, 1180, 1196, 1311, 1341,
 1343, 1349, 1356, 1380, 1387, 1437, 1455, 1466, 1498, 1505
`\cs_set:Npn` 275, 434, 1108, 1114

`\cs_set_eq:NN`
 128, 133, 134, 135, 239, 1253, 1254, 1255, 1256,
 1257, 1264, 1267, 1268, 1277, 1278, 1279, 1280, 1281, 1347
`\cs_set_protected:Npe` 240, 242
`\cs_set_protected:Npn` ... 1237, 1246, 1275, 1296, 1302
`\cs_undefine:N` 136, 438

ctex commands:

`\ctex_after_end_preamble:n` 71
`\ctex_at_end_package:nn` 85
`\ctex_at_end_preamble:n` 70
`\ctex_disable_package:n` 20, 40, 43
`\ctex_if_format_at_least:nTF` 41
`\ctex_replace_package:nn` 42

D

Default 368
`\defaultCJKfontfeatures` 8
depth 16

dim commands:

`\dim_case:nn` 1038, 1366
`\dim_case:nnTF` 975
`\dim_compare:nNnTF` 958
`\dim_const:Nn` 1143
`\dim_eval:n` 202
`\dim_gset:Nn` 1143, 1474
`\dim_if_exist:NTF` 1142
`\dim_max:nn` 1509
`\dim_new:N` 51, 1465
`\dim_set:Nn` 1502, 1507
`\dim_use:N` 223
`\c_zero_dim` 959, 1510

E**else commands:**

`\else:` 145, 255, 260, 308, 316, 1120, 1137
EmboldenFactor 4
EnvCS 4
EnvCS+ 4
EnvCS- 4

exp commands:

`\exp_after:wN` 139,
 234, 252, 253, 258, 259, 261, 262, 273, 301, 1443, 1450, 1459
`\exp_args:Nc` 343, 1158
`\exp_args:Ncc` 1110
`\exp_args:NNNo` 218, 222, 225
`\exp_last_unbraced:Ne` 113
`\exp_not:N` 48, 241, 243, 257, 696, 1410, 1412
`\exp_not:n` 241, 243, 415,
 416, 455, 456, 457, 458, 459, 460, 654, 660, 674, 684, 693
`\exp_stop_f:` 144, 315
experiment/halfright-prebreakpenalty 6
experiment/punct-measure-fix 6

F

FallBack 9

fi commands:

`\fi:` 145, 234, 263, 264, 308, 316, 1120, 1137

format 16
 fp commands:
 \fp_compare:nNnTF 371
 \fp_eval:n 203
 FullLeft 368
 FullRight 368

G

gap 16
 group commands:
 \group_align_safe_begin: 245, 1320, 1373
 \group_align_safe_end: 241, 243, 1351, 1376, 1382
 \group_begin: 214, 274, 1107
 \c_group_begin_token 721
 \group_end: 225, 302, 1130
 \c_group_end_token 727

H

HalfLeft 387
 HalfRight 387
 HangulJamo 387
 HangulJamoL 387
 HangulJamoT 387
 HangulJamoV 387

hbox commands:

 \hbox_set:Nw 215
 \hbox_set_end: 218, 222
 height 16
 hidden 16

I**if commands:**

 \if_catcode:w 256
 \if_cs_exist:w 305
 \if_dim:w 1133
 \if_int_compare:w 231, 314, 1119
 \if_meaning:w 250
 \IfBooleanT 530
 InlineEnv 4
 InlineEnv+ 4
 InlineEnv- 4

int commands:

 \int_compare:nNnTF 149, 162, 182, 374
 \int_const:Nn 68, 356, 379, 385, 1116
 \int_div_truncate:nn 205
 \int_eval:n 280, 287, 297, 311
 \int_gincr:N 1141, 1148
 \int_gset:Nn 210, 1154
 \int_gset_eq:NN 881, 914
 \int_if_exist:NTF 340, 353
 \int_incr:N 575, 585
 \int_max:nn 565
 \int_min:nn 564
 \int_new:N 49, 208, 235, 236, 1153
 \int_set:Nn 560, 564, 565, 570, 571, 581, 590
 \int_set_eq:NN 561
 \int_until_do:nNnn 24

 \int_use:N 370
 \int_value:w 206
 \int_zero:N 724
 \c_one_int 328, 374
 \c_zero_int 65, 330, 1469

K

KaiMingPunct 5
 KaiMingPunct+ 5
 KaiMingPunct- 5

keys commands:

 \keys_define:nn 320, 1233, 1242, 1290
 \keys_set:nn 610, 612

L

LatinPunct 5
 LoadFandol 7
 LocalConfig 3
 LongPunct 5
 LongPunct+ 5
 LongPunct- 5

M

\makeXeCJKactive 323, 327
 \makeXeCJKinactive 324, 327
 Mapping 7
 MiddlePunct 5
 MiddlePunct+ 5
 MiddlePunct- 5

mode commands:

 \mode_if_inner:TF 1468

msg commands:

 \msg_critical:nn 10
 \msg_critical:nnn 20
 \msg_error:nn 57
 \msg_error:nnn 36, 58
 \msg_info:nnnn 63
 \msg_new:nnn 3, 11, 28, 55
 \msg_new:nnnn 56
 \msg_warning:nn 59
 \msg_warning:nnn 60
 \msg_warning:nnnn 61
 \msg_warning:nnnnn 62

N

\newCJKfontfamily 8
 \NewDocumentCommand 44, 99, 327, 329, 527, 551, 594, 599, 616
 NewLineCS 4
 NewLineCS+ 4
 NewLineCS- 4
 \newXeTeXintercharclass 343
 NoBreakCS 6
 NoBreakCS+ 6
 NoBreakCS- 6
 NoBreakLongPunct 5
 NoBreakLongPunct+ 5
 NoBreakLongPunct- 5

NormalSpace 387
 \normalspacedchars 12, 594

P

peek commands:

\peek_after:Nw 246, 252
 \peek_meaning_remove:NTF 882, 915
 \l_peek_token . 250, 257, 888, 921, 1323, 1328, 1331, 1374

PlainEquation 4

PoZheHao 387

PoZheHaoLigature 5

prg commands:

\prg_do_nothing: 128, 130, 267, 268, 650, 696, 1065, 1227
 \prg_new_conditional:Npnn 22, 142, 303, 312, 1117, 1131
 \prg_return_false: 25, 145, 308, 316, 1120, 1137
 \prg_return_true: 25, 145, 308, 316, 1120, 1137

PunctBoundWidth 6

PunctFamily 4

PunctStyle 4

PunctWidth 6

Q

quark commands:

\q_stop 273, 275, 279, 285, 296, 301

R

\RequirePackage 21, 45, 46

reverse commands:

\reverse_if:N 231

RubberPunctSkip 6

S

scan commands:

\scan_stop: 23, 26, 215, 239, 319, 1328, 1346, 1347
 \s_stop 114, 116

sep 16

seq commands:

\seq_gput_right:Nn 346, 347, 358, 707
 \seq_gset_eq:NN 700
 \seq_gset_from_clist:Nn 702
 \seq_map_inline:Nn 730, 743, 756, 778, 808, 1405
 \seq_new:N 336, 337, 699, 701, 704
 \setCJKfallbackfamilyfont 9, 10
 \setCJKfamilyfont 7
 \setCJKmainfont 7
 \setCJKmathfont 9
 \setCJKmonofont 7
 \setCJKsansfont 7

skip 15

skip commands:

\skip_const:Nn 1150
 \skip_gset:Nn 1150
 \skip_horizontal:N 850, 853, 863,
 870, 871, 873, 947, 1007, 1030, 1047, 1049, 1062, 1076,
 1091, 1094, 1097, 1098, 1103, 1185, 1188, 1191, 1210, 1212
 \skip_if_eq:nnTF 151, 160, 168, 185, 867, 1083
 \skip_if_exist:NTF 1149

\skip_if_finite:nTF 956

\skip_new:N 53, 876, 1241, 1288

\skip_set:Nn 219, 223, 226

\skip_set_eq:NN 860, 955, 1070

\skip_use:N 219, 226

\c_zero_skip 151, 160, 168, 185

SlantFactor 4

\SplitArgument 552

str commands:

\str_const:Nn 369

\str_if_eq:nnTF ... 17, 543, 732, 736, 745, 758, 780, 810

subtract 16

symbol 16

sys commands:

\sys_if_engine_xetex:TF 10

T

TeX and L^AT_EX₂_ε commands:

\@begindvi 22, 90, 94, 97

\@empty 22

\@ifpackagelater 19

\@pkgextension 27

\[..... 4

\addCJKfontfeatures 8

\AtBeginDvi 22

\AtBeginShipout 18, 22

\baselineskip 3

\begin 4, 17

\begingroup 36

\catcode 18, 25

\char 19

\CJKecglue 40

\CJKfamily 8, 18

\CJKfamilydefault 9, 10, 12

\CJKfontspec 8

\CJKglue 18, 39

\CJKrmdefault 9

\CJKsout 16, 17, 19

\CJKsymbol 18

\CJKKunderanyline 16

\CJKKunderanysymbol 16, 17

\CJKKunderdblline 16, 19

\CJKKunderdot 15–17, 19

\CJKKunderline 15, 16, 19

\CJKKunderwave 16, 19

\CJKxout 19

\color 16

\colorbox 19, 42

\cprotect 18, 19

\defaultCJKfontfeatures 8

\document 21

\dotfill 40

\emph 19

\end 4, 17

\endgroup 36

\familydefault 9

\footnote	6, 18	\xeCJKEditPunctStyle	13
\footnotemark	6	\xeCJKfntefbox	16, 17
\hbox	17, 42, 51	\xeCJKfntefon	17
\hrulefill	40	\xeCJKnobreak	6, 18
\hskip	3	\xeCJKOffVerbAddon	18
\hyperref	19, 20	\xeCJKResetCharClass	5
\icprotect	18	\xeCJKRestoreSubCJKBLOCK	12
\kern	6	\xeCJKsetkern	13–15
\leaders	40	\xeCJKsetup	2, 3, 13, 15–17
\maxdimen	13, 14	\xeCJKsetwidth	6, 12, 13
\mbox	19, 42	\xeCJKShipoutHook	18
\meaning	25	\xeCJKVerbAddon	7, 18
\newCJKfontfamily	4, 8	\XeTeXdashbreakstate	38
\nobreak	6	\XeTeXinterchartoks	19, 24
\normalfont	9	tex commands:	
\normalspacedchars	12	\tex_font:D	139, 144, 153, 154, 155, 165, 172, 173, 178, 179, 190
\outer	18, 19	\tex_fontdimen:D	153, 154, 155, 165, 172, 173, 178, 179, 190
\par	4, 6	\tex_glueshrink:D	196, 959
\parfillskip	6	\tex_gluestretch:D	195
\path	3	\tex_hskip:D	1230, 1231
\ref	19, 20	\tex_iffontchar:D	144
\relax	19, 48	\tex_ignorespaces:D	1453, 1462
\reset@color	42	\tex_italiccorrection:D	882, 884, 915, 917
\rmfamily	7, 8	\tex_kern:D	1167, 1168, 1494
\setCJKfallbackfamilyfont	9, 10	\tex_lastkern:D	975, 1038, 1135, 1366
\setCJKfamilyfont	4, 7, 8	\tex_lastnodetype:D	1119
\setCJKmainfont	2, 7	\tex_lastskip:D	219, 860, 955, 1070, 1136, 1474
\setCJKmathfont	9	\tex_nullfont:D	141
\setCJKmonofont	7	\tex_penalty:D	65, 67, 1061, 1469
\setCJKsansfont	7	\tex_romannumeral:D	254
\sffamily	7, 8	\tex_spacefactor:D	881, 914
\shipout	22	\tex_spaceskip:D	151, 157, 160, 183, 189, 195, 196
\spacefactor	24	\tex_the:D	139, 646
\textbf	19	\tex_unkern:D	1173
\textcolor	19	\tex_unskip:D	861, 961, 1071, 1176
\textit	19	\tex_XeTeXcharclass:D	271, 374, 584, 592, 597
\textnormal	9	\tex_XeTeXdashbreakstate:D	724
\textrm	7, 8	\tex_XeTeXinterchartokenstate:D	328, 330
\textsf	7, 8	\tex_XeTeXinterchartoks:D	640, 646, 694
\texttt	7, 8	\tex_XeTeXrevision:D	370
\ttfamily	7, 8	\tex_XeTeXversion:D	370
\uline	17	\tex_xspaceskip:D	168, 176, 185, 193
\ULon	17	textformat	
\ULthickness	16		16
\unskip	6	thickness	
\verb	7		16
\verbatim@font	7	tl commands:	
\xeCJK@document@hook	69, 72	\c_space_tl	1287
\xeCJK@document@left@hook	70, 74	\tl_const:Nn	27, 96, 147, 708
\xeCJK@document@right@hook	71, 76	\tl_gclear:N	1019, 1022, 1026, 1178
\xeCJK@first@begindvi	90, 91	\tl_gput_right:Nn	79, 81, 83, 95
\xeCJKCancelSubCJKBLOCK	12	\tl_gset:Nn	1159
\xeCJKchar	19	\tl_if_blank:nTF	560
\xeCJKDeclareCharClass	12	\tl_if_blank_p:n	557, 558
\xeCJKDeclarePunctStyle	4, 13	\tl_if_empty:NTF	667, 671, 680
\xeCJKDeclareSubCJKBLOCK	11, 12	\tl_if_empty:nTF	277, 283, 289, 297
		\tl_if_eq:NNTF	1063

`\tl_if_eq:NnTF` 1018,
 1021, 1024, 1184, 1187, 1190, 1200, 1203, 1206, 1209, 1211
`\tl_if_exist:NnTF` 24
`\tl_if_head_is_N_type:nTF` 120
`\tl_if_novalue:nTF` 564, 565
`\tl_if_single:nTF` 118
`\tl_map_inline:nn` 596
`\tl_new:N` 47, 48, 86, 87, 88, 109, 1155
`\tl_put_right:Nn` 90, 108
`\tl_replace_all:Nnn` 682
`\tl_set:Nn` 665, 669, 678
`\tl_trim_spaces:n` 114, 121, 124
`\tl_use:N` 73, 75, 77, 104

token commands:

`\c_math_toggle_token` 1313, 1358
`\c_space_token` 250
`\token_if_eq_meaning:NNTF` 1328, 1346
`\token_if_group_end:NnTF` 1331
`\token_if_letter:NnTF` 568
`\token_if_macro:NnTF` 1323, 1374
`\token_if_space:NnTF` 888, 921
`\token_to_meaning:N` 273, 301
`\TrimSpaces` 527

U

use commands:

`\use:N` 526, 709, 1136, 1162, 1164
`\use_i:nn` 848
`\use_none:n` 230, 233, 234
`\use_none:nn` 176, 193

V

Verb 7

W

WidowPenalty 3

X

xCJKecglue 3, 1242

xeCJK commands:

`\xeCJK_add_to_shipout:n` 107, 107
`\xeCJK_allow_break:` 64, 64
`\xeCJK_app_inter_class_toks:nnn`
 657, 657, 662, 834, 1416
`\xeCJK_Boundary_and_Default:` 833, 837, 837
`\xeCJK_Boundary_and_FullLeft:N` 1430
`\xeCJK_Boundary_and_FullRight:N` 1432
`\xeCJK_Boundary_and_NormalSp:` 896, 897, 897
`\xeCJK_check_for_ecglue:` .. 838, 857, 1256, 1267, 1280
`\xeCJK_check_for_ecglue_normalsp:`
 898, 910, 1258, 1269, 1282
`\xeCJK_check_for_glue:` 928, 935, 935
`\xeCJK_check_for_xglue:` . 1056, 1065, 1255, 1266, 1279
`\xeCJK_CJK_and_Boundary:w` 1310, 1311, 1311
`\xeCJK_CJK_and_CJK:N` 1386, 1387, 1387
`\xeCJK_class_group_begin:` 719, 719, 821, 929

`\xeCJK_class_group_end:`
 .. 719, 727, 828, 1316, 1317, 1352, 1424, 1442, 1449, 1458
`\xeCJK_class_num:n`
 . 525, 525, 546, 549, 590, 597, 640, 641, 646, 647, 695, 696
`\xeCJK_clear_Boundary_and_CJK_toks:`
 687, 687, 691, 693, 824, 930
`\xeCJK_clear_inter_class_toks:nn`
 649, 649, 672, 770, 771, 772, 773, 774, 823
`\xeCJK_copy_inter_class_toks:nnnn` 663, 663, 733,
 735, 737, 747, 748, 760, 761, 768, 782, 783, 786, 812, 813
`\xeCJK_cs_clear:N` 127, 127, 1265, 1266
`\xeCJK_cs_gc_clear:N` 127, 129
`\xeCJK_declare_char_class:nN` 532, 535,
 537, 605, 606, 607, 608, 626, 627, 628, 629, 630, 631, 633
`\xeCJK_declare_char_class:nn` 529, 532, 532
`\xeCJK_declare_glue_node:n` 1146, 1223
`\xeCJK_declare_node:n`
 1139, 1139, 1218, 1219, 1220, 1221, 1222, 1224
`\xeCJK_fallback_symbol:NN` 794, 825, 932, 1390, 1396, 1402
`\xeCJK_font_gset_to_current:N` 138, 138
`\xeCJK_FullLeft_and_Boundary:` 1434, 1437, 1437
`\xeCJK_FullLeft_and_CJK:` 1395
`\xeCJK_FullRight_and_Boundary:` 1436, 1455, 1455
`\xeCJK_FullRight_and_CJK:` 1401
`\xeCJK_FullRight_and_CJStarter:` 793, 798
`\xeCJK_get_inter_class_toks:nn`
 644, 644, 654, 660, 666, 670, 679
`\xeCJK_glue_to_skip:nN` 212, 212, 1238, 1247, 1276, 1353
`\xeCJK_glyph_if_exist:N` 142
`\xeCJK_glyph_if_exist:NnTF` 142
`\xeCJK_glyph_if_exist_p:N` 142
`\xeCJK_if_CJK_class:N` 303
`\xeCJK_if_CJK_class:NnTF` 303
`\xeCJK_if_CJK_class_p:N` 303
`\xeCJK_if_last_node:n` 1131
`\xeCJK_if_last_node:nTF` 847,
 849, 852, 862, 865, 907, 1072, 1075, 1087, 1090, 1093, 1131
`\xeCJK_if_last_node_p:n` 1131
`\xeCJK_if_last_punct:TF` 1054
`\xeCJK_if_package_loaded:n` 22
`\xeCJK_if_package_loaded:nTF` 22, 35
`\xeCJK_if_package_loaded_p:n` 22
`\xeCJK_if_same_class:NN` 312
`\xeCJK_if_same_class:NnTF` 312
`\xeCJK_if_same_class_p:NN` 312
`\xeCJK_ignore_spaces:w` 1356, 1356
`\xeCJK_int_until_do:nn` 228, 228, 572, 582
`\xeCJK_inter_class_toks:nnn`
 638, 638, 643, 650, 653, 659,
 674, 683, 791, 819, 828, 832, 879, 895, 912, 926, 1310,
 1385, 1393, 1399, 1409, 1411, 1421, 1429, 1431, 1433, 1435
`\xeCJK_make_boundary:` 318, 318
`\xeCJK_make_node:n`
 885, 890, 918, 923, 1139, 1156, 1354, 1369
`\xeCJK_make_space_node:`
 889, 922, 1227, 1227, 1254, 1265, 1278, 1371

- \xeCJK_new_class:n 338, 338, 382, 383, 384, 387, 388, 389, 390, 393, 394, 395, 396, 397, 398
- \xeCJK_no_break: . 64, 66, 788, 790, 801, 1419, 1426, 1444
- \xeCJK_peek_catcode_ignore_spaces:NTF 237, 237, 1313, 1358
- \xeCJK_pre_inter_class_toks:nnn 651, 651, 656, 788, 789, 893, 1485
- \xeCJK_punct_node:N 1443, 1450, 1459, 1498, 1498
- \xeCJK_remove_node: 850, 853, 863, 870, 908, 947, 978, 980, 982, 1041, 1043, 1045, 1047, 1049, 1073, 1076, 1088, 1091, 1094, 1170, 1185, 1188, 1191, 1201, 1204, 1207, 1210, 1212, 1369, 1371
- \xeCJK_replace_inter_class_toks:nnnn 676, 676
- \xeCJK_reset_space_factor: 208, 209, 211, 723
- \xeCJK_save_class:nn 351, 351, 368, 372, 373, 376, 377, 378
- \xeCJK_select_font: 822, 931
- \xeCJK_set_char_class:nnm 545, 549, 578, 578
- \xeCJK_space_glue: 1264, 1287, 1297
- \xeCJK_space_or_xecglue: 908, 1253, 1264, 1277, 1316, 1361, 1383
- \c_xeCJK_space_skip_tl 147, 869, 1085
- \xeCJK_swap_cs:NN 131, 131
- \xeCJK_tl_remove_outer_braces:n 111, 111, 122
- \xeCJK_token_value_charcode:N 271, 272, 272
- \xeCJK_token_value_class:N 270, 270, 306, 314, 315
- \xeCJK_widow_penalty: 982, 1027, 1045, 1207
- xeCJK internal commands:
- _xeCJK_add_special_punct:nn 436
- _xeCJK_after_end_preamble:n 69, 82
- \g_xeCJK_after_end_preamble_hook_tl 77, 83, 88
- _xeCJK_after_preamble:n 69, 80, 89
- \g_xeCJK_after_preamble_hook_tl 73, 81, 87
- \l_xeCJK_aligni_tl 1063
- _xeCJK_at_end_preamble:n 69, 78
- \g_xeCJK_at_end_preamble_hook_tl 75, 79, 86
- \g_xeCJK_base_class_seq 699
- \l_xeCJK_begin_int 235, 570, 572, 574, 575, 580, 582, 584, 585
- _xeCJK_boundary_group_end:n 1325, 1334, 1336, 1342, 1345, 1349
- _xeCJK_boundary_reserve_space: 1324, 1341
- _xeCJK_ccglue_or_space: 980, 1022, 1043, 1073, 1088, 1105, 1204, 1296, 1302
- \l_xeCJK_ccglue_skip 1238, 1241
- _xeCJK_check_for_ecglue: 843, 845, 857, 1267
- _xeCJK_check_for_ecglue_normalsp: 903, 905, 911, 1270
- _xeCJK_check_for_glue_auxi: 938, 1036
- _xeCJK_check_for_glue_auxii: 948, 1008, 1014, 1031, 1052
- _xeCJK_check_for_glue_auxiii: 1055, 1058
- _xeCJK_check_for_glue_skip: 941, 953
- _xeCJK_check_for_glue_skip_consume_and_fallback: 966, 968, 1010
- _xeCJK_check_for_glue_skip_hlist_aux: 993, 999, 1016
- _xeCJK_check_for_glue_skip_kern: 963, 970
- _xeCJK_check_for_glue_skip_nonkern: 964, 988
- _xeCJK_check_for_glue_skip_restore: 984, 986, 1000, 1002, 1005
- _xeCJK_check_for_xecglue: 839, 1256, 1280
- _xeCJK_check_for_xecglue_normalsp: 899, 1259, 1283
- _xeCJK_check_for_xglue: 1066, 1255, 1279
- _xeCJK_check_for_xglue_aux: 1077, 1081
- _xeCJK_check_num_range:nnNN 554, 554, 580
- \c_xeCJK_CJ_chars_clist 443, 633
- \l_xeCJK_CJ_strict_bool 632, 636
- _xeCJK_CJK_and_Boundary_relax:N 1329, 1343
- \c_xeCJK_CJK_chars_clist 462, 626
- \g_xeCJK_CJK_class_seq 699
- _xeCJK_CJK_class_tl:n 306, 310, 709
- \l_xeCJK_CJK_group_bool 319, 722, 726
- \g_xeCJK_CJK_range_clist 618
- \g_xeCJK_CJStarter_range_clist 625
- \c_xeCJK_CL_chars_clist 418, 455
- \c_xeCJK_class_begin_int 379, 385
- _xeCJK_class_csname:n 340, 344, 348, 353, 356, 361, 361, 362, 526, 709
- \g_xeCJK_class_seq 336, 346, 358, 700, 730, 743, 756, 778, 808
- \c_xeCJK_CM_chars_clist 512, 628
- \g_xeCJK_CM_range_clist 620
- \l_xeCJK_ecglue_skip 850, 853, 863, 870, 947, 1047, 1049, 1076, 1091, 1094, 1097, 1185, 1188, 1191, 1210, 1212, 1247, 1276, 1288, 1353
- \l_xeCJK_end_int 236, 571, 572, 580, 582
- _xeCJK_error:n 55, 57
- _xeCJK_error:nn 55, 58, 341, 354
- \c_xeCJK_EX_chars_clist 440, 457
- _xeCJK_font_gset_to_current_aux:NN 139, 140
- \c_xeCJK_FullLeft_chars_clist 413, 607
- \g_xeCJK_FullLeft_range_clist 603
- \c_xeCJK_FullRight_chars_clist 453, 608
- \g_xeCJK_FullRight_range_clist 604
- _xeCJK_get_charcode:w 273, 279, 285, 296
- \g_xeCJK_glue_check_pending_bool 972, 974, 990, 991, 1012, 1013, 1225, 1333
- _xeCJK_glue_node:n 1163, 1230, 1231
- \c_xeCJK_HalfLeft_chars_clist 399, 605
- \g_xeCJK_HalfLeft_range_clist 601
- \c_xeCJK_HalfRight_chars_clist 399, 606
- \l_xeCJK_halfright_nobreak_bool .. 1415, 1418, 1425
- \g_xeCJK_HalfRight_range_clist 602
- \g_xeCJK_HangulJamo_range_clist 621
- \c_xeCJK_HangulJamoL_chars_clist 519, 629
- \g_xeCJK_HangulJamoL_range_clist 622
- \c_xeCJK_HangulJamoT_chars_clist 519, 631
- \g_xeCJK_HangulJamoT_range_clist 624
- \c_xeCJK_HangulJamoV_chars_clist 519, 630
- \g_xeCJK_HangulJamoV_range_clist 623
- \c_xeCJK_hyphens_chars_clist 426, 437, 460

_xeCJK_if_last_glue:TF	216, 841, 901, 940, 1068, 1107, 1175
_xeCJK_if_last_hlist:TF	992, 1107
_xeCJK_if_last_kern:TF	937, 962, 1107, 1172
_xeCJK_if_last_math:TF	946, 1096, 1107
_xeCJK_if_last_none:TF	1107
_xeCJK_if_last_penalty:TF	1107
_xeCJK_if_last_whatsit:TF	943, 998, 1182, 1198
_xeCJK_info:nnn	55, 63
_xeCJK_int_until_do:wn	228, 230, 233, 234
\c_xeCJK_IS_chars_clist	442, 458
\c_xeCJK_iteration_marks_chars_clist	428
\g_xeCJK_last_node_tl	1018, 1019, 1021, 1022, 1024, 1026, 1155, 1159, 1178, 1184, 1187, 1190, 1200, 1203, 1206, 1209, 1211
\l_xeCJK_last_penalty_bool	1060
\l_xeCJK_last_penalty_int	1061
\g_xeCJK_last_punct_tl	800, 802, 1423, 1427, 1439, 1441, 1443, 1445, 1450, 1457, 1459, 1460
\l_xeCJK_last_skip	860, 868, 871, 873, 876, 955, 956, 959, 1007, 1030, 1062, 1070, 1084, 1098, 1103
\l_xeCJK_latin_punct_bool	611, 615
\c_xeCJK_left_tl	1063, 1445
_xeCJK_make_node:N	1158, 1165, 1501, 1503
_xeCJK_make_space_node:	1228, 1254, 1278
_xeCJK_msg_new:nn	55, 55, 363
_xeCJK_msg_new:nnn	56
\g_xeCJK_new_class_seq	336, 347
\c_xeCJK_nobreak_penalty_int	67, 68
_xeCJK_nobreak_zero_glue:	1451
_xeCJK_node:n	977, 979, 981, 1040, 1042, 1044, 1046, 1048, 1161, 1368, 1370
\g_xeCJK_node_int ..	1141, 1144, 1148, 1151, 1153, 1154
\g_xeCJK_non_CJK_class_seq	699, 1405
\c_xeCJK_NormalSpace_chars_clist	399, 627
\g_xeCJK_NormalSpace_range_clist	619
\c_xeCJK_NS_chars_clist	426, 456
\c_xeCJK_OP_chars_clist	404, 415
\c_xeCJK_package_ext_tl	24, 27
_xeCJK_package_hook:nn	69, 84
\g_xeCJK_par_guard_bool	1464, 1473, 1486, 1489, 1492, 1495
\g_xeCJK_par_guard_dim	1465, 1474, 1494
_xeCJK_peek_catcode_false:w	242, 262, 268
_xeCJK_peek_catcode_ignore_spaces_branches:w	246, 248, 253
_xeCJK_peek_catcode_true:w	240, 259, 267
\l_xeCJK_peek_ignore_spaces_bool	25, 244, 251, 269, 1315, 1321, 1360, 1364
\l_xeCJK_peek_search_token	239, 257, 266
\c_xeCJK_PO_chars_clist	452, 459
\l_xeCJK_pozhehao_ligature_bool	609, 614
\c_xeCJK_PR_chars_clist	411, 416
_xeCJK_punct_bound_unitization:NN ..	1500, 1505, 1505
_xeCJK_punct_boundary_guard: ..	1446, 1461, 1464, 1466
_xeCJK_punct_glue:NN	802, 1427, 1445, 1460
_xeCJK_punct_if_middle:NTF	1439
\g_xeCJK_punct_measure_fix_bool	1471
_xeCJK_punct_rule:NN	800, 1423, 1441, 1457
_xeCJK_recover_ecglue_whatsit:	1180
_xeCJK_recover_glue_whatsit:	944, 1196
_xeCJK_replace_space:	842, 858, 858, 902
_xeCJK_reserve_space_aux:	1375, 1380
\l_xeCJK_reserve_space_bool	1295, 1301, 1309
\g_xeCJK_reset_color_pending_bool ..	995, 997, 1226
\c_xeCJK_right_tl ..	800, 802, 1423, 1427, 1441, 1457, 1460
_xeCJK_save_CJK_class:n	705, 718
_xeCJK_select_font:	803
_xeCJK_set_char_class_aux:Nnw	532, 545, 551
_xeCJK_set_char_class_eq:nn	588, 588
\l_xeCJK_shipout_hook_bool	101, 103, 110
\l_xeCJK_shipout_hook_tl	104, 108, 109
\g_xeCJK_space_factor_int	149, 162, 182, 203, 206, 208, 881, 914
_xeCJK_space_skip_scale:nnn ..	164, 170, 183, 187, 200
_xeCJK_swap_cs_aux:w	133, 135, 136
_xeCJK_tl_remove_outer_braces:w	114, 116
_xeCJK_tmp:nn	1108, 1123, 1124, 1125, 1126, 1127, 1128, 1129
_xeCJK_tmp:w	275, 301, 434, 437, 438
_xeCJK_tmp_aux:NNn	1110, 1114
\l_xeCJK_tmp_bool	52
\l_xeCJK_tmp_box	50, 215, 223
\l_xeCJK_tmp_clist	54, 534, 535
\l_xeCJK_tmp_dim	51, 1500, 1501, 1502, 1503
\l_xeCJK_tmp_int	49, 362, 581, 584, 590, 592
\l_xeCJK_tmp_skip	53
\l_xeCJK_tmp_tl	47, 665, 667, 669, 671, 674, 678, 680, 682, 684
\l_xeCJK_tmpb_tl	48
_xeCJK_update_clear_toks:n	689, 711
_xeCJK_warning:n	59
_xeCJK_warning:nn	55, 60
_xeCJK_warning:nnn	61
_xeCJK_warning:nnnn	62
\l_xeCJK_xecglue_bool	1252, 1263, 1274, 1289
\c_xeCJK_xetex_version_str	369, 371
xeCJKactive	3, 320
\xeCJKCancelSubCJKBLOCK	12
\xeCJKDeclareCharClass	12, 527
\xeCJKDeclarePunctStyle	13
\xeCJKDeclareSubCJKBLOCK	11
\xeCJKEditPunctStyle	13
\xeCJKfntefon	17
\xeCJKnobreak	18
\xeCJKOffVerbAddon	18
\xeCJKResetCharClass	12, 616, 637
\xeCJKResetPunctClass	12, 530, 599, 634
\xeCJKRestoreSubCJKBLOCK	12
\xeCJKsetkern	13
\xeCJKsetup	2
\xeCJKsetwidth	12
\xeCJKShipoutHook	18, 89
\xeCJKVerbAddon	18