

xpinyin 宏包

李清* 黄晨成†

2026/09/27 v3.2‡

第 1 节 简介

xpinyin 是一个 L^AT_EX 宏包,提供了为汉字自动注音的功能。

宏包最初由李清编写和维护,并纳入 ctex-kit 项目管理。自 2022 年至 2026 年,社区已与李清失联逾四年。考虑到这段时间陆续有缺陷报告和功能请求,与 CTAN 联络后,社区决定接手维护该宏包。自 2026 年 6 月起,由黄晨成实际维护。

第 2 节 基本用法

xpinyin 支持采用 GBK 和 UTF-8 编码的 T_EX 源文件,建议总是使用 UTF-8。如果使用 L^AT_EX 或 pdfL^AT_EX 的编译方式,则根据编码的情况,xpinyin 依赖 CJK 或者 CJKutf8 宏包。如果使用 XeL^AT_EX,则依赖 xeCJK 宏包。如果它们没有在 xpinyin 之前被载入,xpinyin 将根据编译方式自动选择,L^AT_EX 或 pdfL^AT_EX 将使用 CJKutf8。

xpinyin 还依赖 l3kernel 和 l3packages,使用 (pdf)L^AT_EX 下的 GBK 编码时,还将依赖 xCJK2uni。

需要注意的是,xpinyin 缺省将拼音的字体设置为与文档的主字体(\normalfont)相同,所以为了保证声调字母的正确输出,应该选用合适的西文主字体。也可以通过将在下一节介绍的 选项来单独设置拼音的字体。

X_{La}T_EX 下的简单示例:

```
\documentclass{article}
\usepackage{xeCJK}
\usepackage{xpinyin}
\setmainfont{CMU Serif}
\setCJKmainfont{SimSun}

\begin{document}
\xpinyin*{汉语拼音示例}
\end{document}
```

(pdf)L^AT_EX 下的简单示例:

```
\documentclass{article}
\usepackage{CJKutf8}
\usepackage{xpinyin}
\usepackage[T1]{fontenc}
\usepackage{lmodern}

\begin{document}
\begin{CJK}{UTF8}{gbsn}
\xpinyin*{汉语拼音示例}
\end{CJK}
\end{document}
```

运行上述示例要求系统安装了设置的字体,源文件用 UTF-8 编码保存,使用相应的编译方式。

xpinyin 可以与 ctex 宏包或文档类共同使用,使用方式与上面类似。

*sobenlee@gmail.com

†liamhuang0205@gmail.com

‡ctex-kit rev. cc4b4d85.

第3节 用户手册

pinyininscope `\begin{pinyininscope}[(options)]`

.....
`\end{pinyininscope}`

为 pinyininscope 环境中的汉字自动注音。例如

```
1 \begin{pinyininscope}
2 列位看官：你道此书从何而来？说起根由，虽近荒唐，细按则深有趣味。
3 待在下将此来历注明，方使阅者\xpinyin{了}{liao3}然不惑。
4 \end{pinyininscope}
```

liè wèi kàn guān nǐ dào cǐ shū cóng hé ér lái shuō qǐ gēn yóu suī jìn huāng táng xì àn zé shēn yǒu qù wèi dài zài xià jiāng cǐ lái lì zhù míng
 列位看官：你道此书从何而来？说起根由，虽近荒唐，细按则深有趣味。待在下将此来历注明，
 fāng shǐ yuē zhè liǎo rán bù huò
 方使阅者了然不惑。

可选项 `(options)` 用于局部设置拼音的格式，将在下面说明。

\xpinyin `\xpinyin [(options)] {(单个汉字)} {(拼音)}`
`\xpinyin* [(options)] {(文字)}`

对于多音字，可以使用 `\xpinyin` 为其设置拼音；而 `\xpinyin*` 相当于 pinyininscope 环境的命令形式。`\xpinyin` 可以在 pinyininscope 环境和 `\xpinyin*` 中使用。例如，

cháng
 长
 zhēn shì yīn mèng huàn shí tōng líng
 甄士隐梦幻识通灵
 zhòng yào
 重要

```
1 \xpinyin{长}{chang2}\\
2 \xpinyin*{甄士隐梦幻识通灵}\\
3 \xpinyin*{\xpinyin{重}{zhong4}}要}
```

\pinyin `\pinyin [(options)] {(拼音)}`

用于输出拼音。声调用音节末尾的数字 1-5 表示(5 是轻声，不加调号)。为了输入的方便 ü 可以用 v 代替；两种写法都可以，结果相同。

lǚ zi

nǚ hái zi

lǚ zi

nǚ hái zi

```
1 \pinyin{lv2zi}\\
2 \pinyin{nv3hai2zi}\\
3 \pinyin{lü2zi}\\
4 \pinyin{nü3hai2zi}
```

v 只在 l、n 之后才当作 ü：ju、qu、xu、yu 按《汉语拼音方案》本就不写分音符，没有歧义。直接写 ü 时同样按这条规则处理，所以 `\pinyin{jü4}` 与 `\pinyin{jv4}` 都得到 jù。

jù

jù

```
1 \pinyin{jv4}\\
2 \pinyin{jü4}
```

整段输入里没有声调数字时原样输出，不做任何替换：`\pinyin{nü}` 得到 nü，`\pinyin{nv}` 得到 nv。

全大写的音节排不出调号(`\pinyin{NV3}` 得到字面的 NV3)，这是原有限制：判断调号位置用的表里只有小写元音。首字母大写不受影响，`\pinyin{Nü3}` 与 `\pinyin{Nv3}` 都得到 Nǚ。

\setpinyin `\setpinyin {(汉字)} {(拼音)}`

xpinyin 宏包的拼音数据 (xpinyin-database.def) 来源于 Unicode 的 UniHan 数据库¹中的 UniHan_Readings.txt 文件。对于多音字，一般来说这个文件选用的是常用读音。可以使用 `\setpinyin` 来设置多音字的首选读音。另外，“子”“們”“啊”“啦”“哇”这几个主要用作轻声助词或后缀的字，本宏包把首选读音固定为轻声(如“子”为 zi)，不随 UniHan 数据的更新而改变。使用了 query 选项时，`\setpinyin` 同时更新查询命令使用的数据，注音与查询给出的首选读音始终一致。

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

用于在导言区或文档中，设置拼音的格式。目前可以设置的 `<key>` 如下介绍。

¹<http://www.unicode.org/Public/UNIDATA/UniHan.zip>

`ratio ratio = {<number>}`

设置拼音字体大小与当前正文字体大小的比例,缺省值是 0.4。

`vsep vsep = {<dimen>}`

设置拼音的基线与汉字基线的间距,缺省值是 1 em。

`hsep hsep = {<skip>}`

设置注音汉字之间的间距,缺省值与 `\CJKglue` 的值相同。为了断行时行末的对齐,设置的 `<skip>` 最后有一定的弹性。例如

```
1 \xpinyin*[ratio={.7},hsep={.5em plus .1em},vsep={1.1em}]{贾雨村风尘怀闺秀}
```

jǐǎ yǔ cūnfēngchénhuái guī xiù
贾 雨 村 风 尘 怀 闺 秀

`pysep pysep = {<glue>}`

设置 `\pinyin` 输出的相邻两个汉语拼音的空白,缺省值是一个空格。

`font font = {<font>}`

设置拼音的字体,缺省值是 `\normalfont`,即以正文西文字体相同。为了保证拼音能正确输出,最好选用收字量较大的西文字体。

`format format = {<format>}`

设置拼音的其他格式,例如颜色等,缺省值为空。

`multiple multiple = {<format>}`

设置多音字拼音的其他格式,缺省值为空。可以通过这个选项来提醒校正多音字的拼音。注意这个选项会覆盖 `format` 设置的格式。例如本文档设置多音字拼音的颜色是红色(需要载入 `color` 宏包):

```
\xpinyinsetup{multiple={\color{red}}}
```

`footnote footnote = <true|false>`

是否对拼音环境中的脚注(`\footnote`)汉字加上拼音。缺省值为 `false`。更一般的情况,请使用 `\disablepinyin`。

`\disablepinyin` 和 `\enablepinyin` 分别用于在拼音环境 (`pinyinscope`) 中临时取消和恢复对汉字的注音。`\disablepinyin` 还提供带星号的形式,可以把禁用范围扩展到 `\xpinyin` 命令本身。

Updated: 2026-07-13

第 4 节 查询汉字的读音

上面几节的命令都是把拼音排在汉字上方。如果只想知道某个汉字读什么,而不想排出注音,可以用本节的四个查询命令。它们返回一段文字,不排版任何别的东西,因此可以写在其他命令的参数里。用于索引排序键时有一点要注意,见“给索引条目加拼音排序键”一节。

这组命令需要一份额外的数据表,加载宏包时用 `query` 选项要求载入:

```
\usepackage[query]{xpinyin}
```

不写这个选项时, 注音功能照常可用, 但调用查询命令会报错。之所以要分开, 是因为这份表有一兆多字节, 只用注音功能的文档不必为载入它多花时间。

`xpinyin` 本身支持 X_{TeX} 与 pdf_{TeX} 两条路线, 而查询命令只在 X_{TeX} 下提供。 pdf_{TeX} 加 `CJKutf8` 的路线上, 一个汉字由多个字节组成, 取不到本组命令所需的码位, 此时使用 `query` 选项会报错。

(Lua_{TeX} 目前不在本宏包的支持范围内, 加载时就会中止, 与查询命令无关。)

`\xpinyinvalue` `\xpinyinvalue` {<单个汉字>}
 `\xpinyinvalue*` {<单个汉字>}

New: 2026-08-06

返回这个汉字的拼音。带星号的形式返回全部读音, 用逗号分隔。

中的读音是 `zhōng`。

行有几个读音: `háng, hàng, héng, xíng`。

1	中 的读音是 <code>\xpinyinvalue{中}</code> 。\\
2	行 有几个读音: <code>\xpinyinvalue*{行}</code> 。

`\xpinyininitial` `\xpinyininitial` {<单个汉字>}
 `\xpinyininitial*` {<单个汉字>}

New: 2026-08-06

返回拼音的第一个字母。带星号的形式返回全部读音各自的首字母, 重复的只保留一个。

这个命令适合给索引条目自动加分组前缀: 全部汉字的首字母都落在 `a` 到 `z` 之内, 因此可以据此把中文条目分成若干组。

汉归入 `h` 组,

安归入 `a` 组。

行跨越 `h, x` 两组。

1	汉 归入 <code>\xpinyininitial{汉}</code> 组, \\
2	安 归入 <code>\xpinyininitial{安}</code> 组。\\
3	行 跨越 <code>\xpinyininitial*{行}</code> 两组。

`\xpinyinshengmu` `\xpinyinshengmu` {<单个汉字>}
`\xpinyinyunmu` `\xpinyinyunmu` {<单个汉字>}

New: 2026-08-06

分别返回声母与韵母, 两者都提供带星号的形式。零声母的字(“安”“儿”一类)声母为空。

中: 声母 `zh`, 韵母 `ōng`。

安: 声母为空, 韵母 `ān`。

1	中: 声母 <code>\xpinyinshengmu{中}</code> ,
2	韵母 <code>\xpinyinyunmu{中}</code> 。\\
3	安: 声母为空,
4	韵母 <code>\xpinyinyunmu{安}</code> 。

`tone` `tone = mark|number|none|v`

New: 2026-08-06 声调用什么形式表示, 缺省值为 `mark`。四种取值的区别如下:

取值	含义	中	女
<code>mark</code>	调号标在字母上	<code>zhōng</code>	<code>nǚ</code>
<code>number</code>	声调写成数字	<code>zhong1</code>	<code>nv3</code>
<code>none / v</code>	不带声调	<code>zhong</code>	<code>nv</code>

`none` 与 `v` 目前结果相同。保留两个名字是因为“不带声调”和“把 `ü` 写成 `v`”本是两件事, 只是本宏包内部一律用 `v` 记录 `ü`, 所以结果相同。

`number` 形式里 `ü` 写作 `v`, 与 `\pinyin` 的输入约定一致, 所以查询结果可以直接传给 `\pinyin`:

`nv3` 传给 `\pinyin` 得到 `nǚ`。

1	<code>\xpinyinsetup{tone=number}</code>
2	<code>\xpinyinvalue{女}</code> 传给 <code>\tn{pinyin}</code>
3	得到 <code>\pinyin{nv3}</code> 。

scheme `scheme = literal|official`

New: 2026-08-06 声母与韵母按什么规则切分，缺省值为 `literal`。这个选项只影响 `\xpinyinshengmu` 与 `\xpinyinyunmu`，不影响另外两个命令。

两种规则的差别只在 `y` 和 `w`。`literal` 把它们当作声母，切分结果拼回去就是原来的拼音。`official` 依据《汉语拼音方案》，把 `y` 和 `w` 视为介音 `i`、`u`、`ü` 的书写形式，声母为空，并把韵母还原成音位形式。

汉字	拼音	literal		official
		声母	韵母	韵母
王	wáng	w	áng	uang
月	yuè	y	uè	ve
鱼	yú	y	ú	v
安	ān		ān	an

`official` 模式下的韵母不带声调。还原出来的形式可能以 `v` 开头 (`yue` 还原成 `ve`)，而调号只能标在元音上，`\pinyin` 也只在 `l` 与 `n` 之后才把 `v` 当作 `ü`，所以给这种形式标调会排出错误的字形。出于同样的原因，`literal` 模式下以 `v` 开头的韵母 (“女”的韵母) 也不带声调。

`yo` 一类音节按“`y` 后接其他元音则 `y` 变 `i`”这条通则处理成 `io`。这个音节的实际读法在语言学上有不同意见，本宏包不对它做特殊处理。

query `query`

New: 2026-08-06 宏包选项，要求载入查询表。只能在 `\usepackage` 时给出，不能中途用 `\xpinyinsetup` 打开：数据表只在加载宏包时读入一次。

4.1 关于可展开性

查询命令可以放在需要参数展开成字面文字的位置，例如 `bib2gls` 的字段。除了下面两种情况，四个命令都是完全可展开的：

- `tone` 取 `mark` 时。带调号的输出要经过 `\pinyin` 的标调过程，它把内容排版出来而不是返回文字。
- 带星号的形式。它需要去掉重复项，去重时要用到内部的列表变量。

也就是说，需要字面文字时请选 `number`、`none` 或 `v`，并使用不带星号的形式。

4.2 给索引条目加拼音排序键

`\index` 不会展开参数。它先用 `\@sanitize` 把参数变成字符，再原样写进 `.idx`，所以直接写

```
\index{\xpinyinvalue{汉}@汉字}
```

得到的是 `\indexentry{\xpinyinvalue{汉}@汉字}{1}`——排序键里是这串命令本身，而不是 `han`。这与查询命令能否展开无关，是 `\index` 本来的行为。

要用拼音作排序键，得自己先展开一次再交给 `\index`：

```
\newcommand\pyindex[2]{%
  \begingroup
    \xpinyinsetup{tone=none}%
    \edef\pyindexkey{#1}%
    \expandafter\index\expandafter{\pyindexkey @#2}%
  \endgroup}
...
汉字\pyindex{\xpinyinvalue{汉}}{汉字}
安全\pyindex{\xpinyinvalue{安}}{安全}
```

这样写进 `.idx` 的是 `\indexentry{han@汉字}{1}`, 经 `makeindex` 排序后「安全」「汉字」「中国」按 `an`、`han`、`zhong` 的顺序排列。

中间变量请用一个普通的名字。若写成 `\@tempa` 这类含 `@` 的名字, 正文里 `@` 不是字母, `\edef\@tempa{...}` 定义的并不是 `\@tempa`, 排序键会多出一个空格 (`han @汉字`); `makeindex` 按字面比较, 这会让它与手写的 `han@汉字` 分成两个条目。要用这类名字就得把整段放在 `\makeatletter` 与 `\makeatother` 之间。

4.3 多音字

不带星号的形式取首选读音, 与 `\xpinyin` 的取值一致; 哪个读音算首选, 由 `UniHan` 数据库决定, 也可以用 `\setpinyin` 改写——`\setpinyin` 同时修改注音与查询两份数据, 因此“长”经 `\setpinyin{长}{chang2}` 之后, `\xpinyin` 与 `\xpinyinvalue` 给出的都是 `cháng`, 据此算出的排序键也落在 `c` 组。带星号的形式给的是全部读音, 不受 `\setpinyin` 影响。

需要注意的是, 多音字的不同读音常常落在不同的首字母上。数据库里的五千多个多音字中, 约有三分之二属于这种情况。因此按首字母给索引分组时, 要留意条目可能应当出现在两处。带星号的形式给出全部读音, 便于自行判断:

行的全部读音: `háng,hàng,héng,xíng`
涉及的首字母: `h,x`

1	行的全部读音: <code>\xpinyinvalue*{行}\</code>
2	涉及的首字母: <code>\xpinyininitial*{行}</code>

第5节 代码实现

```

1 <*package>
2 <@=@=xpinyin>
3 \msg_new:nnn { xpinyin } { l3-too-old }
4 {
5   Support~package~'expl3'~too-old. \\\
6   Please~update~an~up~to~date~version~of~the~bundles\\\
7   'l3kernel'~and~'l3packages'\\\
8   using~your~TeX~package~manager~or~from~CTAN.
9 }
10 \@ifpackagelater { expl3 } { 2019/03/05 } { }
11 { \msg_error:nn { xpinyin } { l3-too-old } }

12 \msg_new:nnn { xpinyin } { engine-not-supported }
13 { Engine~'\c_sys_engine_str'~is~not~yet~supported,~xpinyin~will~abort! }
14 \bool_lazy_or:nnF
15 { \sys_if_engine_xetex_p: }
16 { \sys_if_engine_pdftex_p: }
17 { \msg_critical:nn { xpinyin } { engine-not-supported } }

```

查询命令需要额外的数据表, 只有加载 `query` 选项时才会读入。未加载时直接报错, 而不是让命令展开成空: 后者会让排序键不加提示地变成空串, 发现问题的位置离出错原因很远。

```

18 \msg_new:nnn { xpinyin } { query-not-loaded }
19 {
20   Pinyin~query~table~is~not~loaded. \\\
21   Load~the~package~with~the~'query'~option:\\\
22   \iow_char:N \usepackage[query]{xpinyin}
23 }
24 \msg_new:nnn { xpinyin } { query-no-reading }
25 { No~pinyin~reading~for~'#1'~in~the~database. }

```

可展开的报错在终端上只显示到行宽为止 (实测 `take exactly...` 之后被截断), 所以关键信息要放在第一行开头, 并且尽量短。完整说明放在第二行, `.log` 里仍能看到完整内容。

```

26 \msg_new:nnn { xpinyin } { query-bad-argument }
27 {
28   Query~one~character~at~a~time~(got~'#1'). \\\
29   Pinyin~query~commands~take~exactly~one~character.
30 }

```

查询命令只在 XeTeX 下提供。取汉字码位要用 `\#1`, 而 pdfTeX 加 CJKutf8 的路线上一个汉字是多个字节, 这个写法拿不到码位; 包内的 `\__xpinyin_to_unicode:n` 依赖 `\CJK@plane`, 需要字符级钩子提供的上下文, 也不能在这里直接调用。

```

31 \msg_new:nnn { xpinyin } { query-needs-unicode }
32 {
33   Pinyin~query~commands~are~only~available~with~XeTeX. \\\
34   They~are~not~available~with~pdfTeX.
35 }

36 \cs_if_exist:NF \NewDocumentCommand
37 { \RequirePackage { xparse } }

```

`\c__xpinyin_tone_prop` 重音标记 `\`、\'` 和 `\=` 在 `tabbing` 环境中被移作他用, 为避免错误, 我们使用内部命令 `\@tabacckludge` 或 `\a` 来定义。

```

38 \prop_const_from_keyval:Nn \c__xpinyin_tone_prop
39 {
40   { ā } = { \@tabacckludge= a } ,
41   { á } = { \@tabacckludge' a } ,
42   { ä } = { \v a } ,
43   { à } = { \@tabacckludge` a } ,
44   { õ } = { \@tabacckludge= o } ,
45   { ó } = { \@tabacckludge' o } ,
46   { ö } = { \v o } ,
47   { ò } = { \@tabacckludge` o } ,
48   { ē } = { \@tabacckludge= e } ,
49   { é } = { \@tabacckludge' e } ,
50   { ë } = { \v e } ,
51   { è } = { \@tabacckludge` e } ,
52   { ū } = { \@tabacckludge= u } ,
53   { ú } = { \@tabacckludge' u } ,
54   { ů } = { \v u } ,
55   { ù } = { \@tabacckludge` u } ,
56   { ṁ } = { \@tabacckludge' m } ,
57   { ṇ } = { \@tabacckludge' n } ,
58   { ṅ } = { \v n } ,
59   { ṅ } = { \@tabacckludge` n } ,
60   { ī } = { \@tabacckludge= { \i } } ,
61   { í } = { \@tabacckludge' { \i } } ,
62   { ĩ } = { \v { \i } } ,
63   { ï } = { \@tabacckludge` { \i } } ,
64   { ü } = { \" u } ,
65   { ũ } = { \@tabacckludge= { \" u } } ,
66   { ú } = { \@tabacckludge' { \" u } } ,
67   { ů } = { \v { \" u } } ,
68   { ù } = { \@tabacckludge` { \" u } }
69 }

```

```

\__xpinyin_UTF_char:nn 70 \cs_new_protected:Npn \__xpinyin_UTF_char:nn #1#2
71 {
72   \cs_if_exist:cF { u8:#1 }
73   { \tl_const:cn { u8:#1 } {#2} }
74 }

```

```

\__xpinyin_GBK_char:nn 75 \cs_new_protected:Npn \__xpinyin_GBK_char:nn #1#2
76 {
77   \__xpinyin_UTF_char:nn {#1} {#2}
78   \exp_args:Nx \__xpinyin_GBK_char_aux:nn { \tl_head:n {#1} } {#1}
79 }
80 \cs_new_protected:Npn \__xpinyin_GBK_char_aux:nn #1#2
81 { \exp_args:Nf \__xpinyin_GBK_char_aux:nnn { \int_eval:n { `#1 } } {#1} {#2} }
82 \cs_new_protected:Npn \__xpinyin_GBK_char_aux:nnn #1#2#3
83 {
84   \cs_if_exist:cF { __xpinyin_UTF_ #1 :w }
85   {
86     \exp_args:Nf \__xpinyin_GBK_char_def:nnn
87     {
88       \int_case:nn { \tl_count:n {#3} }

```

```

89         {
90             { 2 } { ##1 }
91             { 3 } { ##1##2 }
92             { 4 } { ##1##2##3 }
93         }
94     }
95     {#1} {#2}
96     \exp_args:Nc \__xpinyin_save_UTF_cs:Nn { __xpinyin_UTF_ #1 :w } {#1}
97     \tl_gput_right:Nx \c__xpinyin_reset_UTF_catcode_tl
98     { \char_set_catcode:nn {#1} { \char_value_catcode:n {#1} } }
99     \char_set_catcode_active:n {#1}
100 }
101 }
102 \cs_new_protected:Npn \__xpinyin_GBK_char_def:nnn #1#2#3
103 {
104     \cs_new_protected:cpn { __xpinyin_UTF_ #2 :w } #1
105     { \use:c { u8: \tl_to_str:n { #3#1 } } }
106 }
107 \tl_new:N \c__xpinyin_reset_UTF_catcode_tl
\__xpinyin_save_UTF_cs:Nn
108 \group_begin:
109 \char_set_catcode_active:n { 126 }
110 \cs_new_protected:Npn \__xpinyin_save_UTF_cs:Nn #1#2
111 {
112     \group_begin:
113     \char_set_lccode:nn { 126 } {#2}
114     \tex_lowercase:D
115     {
116         \group_end:
117         \tl_gput_right:Nn \c__xpinyin_reset_UTF_cs_tl { \cs_set_eq:NN ~ #1 }
118     }
119 }
120 \group_end:
121 \tl_new:N \c__xpinyin_reset_UTF_cs_tl
122 \bool_new:N \g__xpinyin_GBK_bool
123 \@ifpackageloaded { xeCJK }
124 { \AtEndOfPackage { \__xpinyin_adjust_xeCJK_hook: } }
125 {
126     \@ifpackageloaded { CJKutf8 }
127     {
128         \prop_map_function:NN \c__xpinyin_tone_prop \__xpinyin_UTF_char:nn
129         \AtEndOfPackage { \__xpinyin_adjust_CJK_hook: }
130     }
131     {
132         \@ifpackageloaded { CJK }
133         {
134             \RequirePackage { xCJK2uni }
135             \prop_map_function:NN \c__xpinyin_tone_prop \__xpinyin_GBK_char:nn
136             \AtEndOfPackage
137             {
138                 \tl_put_right:Nn \l__xpinyin_pinyin_box_hook_tl
139                 { \c__xpinyin_reset_UTF_cs_tl }
140                 \__xpinyin_adjust_CJK_hook:
141                 \tl_use:N \c__xpinyin_reset_UTF_catcode_tl
142             }
143             \bool_gset_true:N \g__xpinyin_GBK_bool
144         }
145     }
146     \sys_if_engine_xetex:TF
147     {
148         \RequirePackage { xeCJK }
149         \AtEndOfPackage { \__xpinyin_adjust_xeCJK_hook: }
150     }
151     {
152         \RequirePackage { CJKutf8 }
153         \prop_map_function:NN \c__xpinyin_tone_prop \__xpinyin_UTF_char:nn
154         \AtEndOfPackage { \__xpinyin_adjust_CJK_hook: }

```

```

155     }
156   }
157 }
158 }

\l_xpinyin_tmpa_box 159 \box_new:N \l_xpinyin_tmpa_box
\l_xpinyin_tmpb_box 160 \box_new:N \l_xpinyin_tmpb_box

\__xpinyin_width:Nn 161 \cs_new_protected:Npn \__xpinyin_width:Nn #1#2
162 {
163   \hbox_set:Nn \l_xpinyin_tmpa_box {#2}
164   #1 = \box_wd:N \l_xpinyin_tmpa_box
165 }

\__xpinyin_make_pinyin_box:nnn 166 \cs_new_protected:Npn \__xpinyin_make_pinyin_box:nnn #1#2#3
167 {
168   \__xpinyin_leavevmode:
169   \hbox_overlap_right:n
170   {
171     \hbox_set:Nn \l_xpinyin_tmpa_box
172     { \__xpinyin_CJKsymbol_hook: \__xpinyin_save_CJKsymbol:n {#2} }
173     \hbox_set:Nn \l_xpinyin_tmpb_box
174     {
175       \l_xpinyin_pinyin_box_hook_tl
176       \__xpinyin_select_font:
177       \l_xpinyin_format_tl
178       \clist_if_exist:cT { c__xpinyin_multiple_ #1 _clist }
179       { \l_xpinyin_multiple_tl }
180       {#3}
181     }
182     \dim_compare:nNnT
183     { \box_wd:N \l_xpinyin_tmpb_box } >
184     { \box_wd:N \l_xpinyin_tmpa_box + \l_xpinyin_CJKglue_dim }
185     {
186       \box_resize_to_wd_and_ht:Nnn \l_xpinyin_tmpb_box
187       { \box_wd:N \l_xpinyin_tmpa_box + \l_xpinyin_CJKglue_dim }
188       { \box_ht:N \l_xpinyin_tmpb_box }
189     }
190     \box_move_up:nn { \l_xpinyin_vsep_tl }
191     {
192       \hbox_to_wd:nn { \box_wd:N \l_xpinyin_tmpa_box }
193       { \tex_hss:D \box_use_drop:N \l_xpinyin_tmpb_box \tex_hss:D }
194     }
195   }
196 }
197 \tl_new:N \l_xpinyin_pinyin_box_hook_tl
198 \sys_if_engine_pdftex:T
199 {
200   \tl_put_right:Nn \l_xpinyin_pinyin_box_hook_tl
201   { \cs_set_eq:NN \CJK@plane \tex_undefined:D }
202 }
203 \cs_generate_variant:Nn \__xpinyin_make_pinyin_box:nnn { x }

\__xpinyin_CJKsymbol:n 204 \cs_new_protected:Npn \__xpinyin_CJKsymbol:n #1
205 { \__xpinyin_CJKsymbol:xn { \__xpinyin_to_unicode:n {#1} } {#1} }
206 \cs_new_protected:Npn \__xpinyin_CJKsymbol:nn #1#2
207 {
208   \__xpinyin_make_pinyin_box:nnn {#1} {#2} { \use:c { c__xpinyin_ #1 _tl } }
209   \__xpinyin_save_CJKsymbol:n {#2}
210 }
211 \cs_generate_variant:Nn \__xpinyin_CJKsymbol:nn { x }

pinyininscope 212 \NewDocumentEnvironment { pinyininscope } { 0 { } }
213 {
214   \keys_set:nn { xpinyin } {#1}
215   \enablepinyin
216 }
217 { }
```

```

\xpinyin 218 \NewDocumentCommand \xpinyin { s O { } m }
219 {
220   \mode_leave_vertical:
221   \bool_if:NTF \l__xpinyin_enable_all_bool
222   {
223     \IfBooleanTF {#1}
224     {
225       \group_begin:
226       \keys_set:nn { xpinyin } {#2}
227       \enablepinyin
228       #3
229       \group_end:
230     }
231     {
232       \group_begin:
233       \keys_set:nn { xpinyin } {#2}
234       \bool_if:NF \l__xpinyin_enable_bool
235       { \__xpinyin_width:Nn \l__xpinyin_CJKglue_dim { \CJKglue } }
236       \__xpinyin_single_aux:nn {#3}
237     }
238   }
239   { \IfBooleanTF {#1} {#3} { \use_i:nn {#3} } }
240 }

```

```

\l__xpinyin_enable_bool 241 \bool_new:N \l__xpinyin_enable_bool
\l__xpinyin_enable_all_bool 242 \bool_new:N \l__xpinyin_enable_all_bool
243 \bool_set_true:N \l__xpinyin_enable_all_bool

\__xpinyin_CJKglue: 244 \cs_new_protected:Npn \__xpinyin_CJKglue:
245 { \skip_horizontal:n { \l__xpinyin_hsep_tl } }

```

```

\enablepinyin 246 \NewDocumentCommand \enablepinyin { }
247 {
248   \bool_if:NF \l__xpinyin_enable_all_bool
249   { \bool_set_true:N \l__xpinyin_enable_all_bool }
250   \bool_if:NF \l__xpinyin_enable_bool
251   {
252     \tl_if_empty:NF \l__xpinyin_hsep_tl
253     {
254       \cs_set_eq:NN \__xpinyin_save_CJKglue: \CJKglue
255       \cs_set_eq:NN \CJKglue \__xpinyin_CJKglue:
256     }
257     \__xpinyin_width:Nn \l__xpinyin_CJKglue_dim { \CJKglue }
258     \__xpinyin_replace_CJKsymbol:
259     \__xpinyin_restore_footnote:
260     \bool_set_true:N \l__xpinyin_enable_bool
261   }
262 }

```

```

\disablepinyin 263 \NewDocumentCommand \disablepinyin { s }
264 {
265   \bool_if:NT #1
266   { \bool_set_false:N \l__xpinyin_enable_all_bool }
267   \bool_if:NT \l__xpinyin_enable_bool
268   {
269     \cs_if_eq:NNT \CJKglue \__xpinyin_CJKglue:
270     { \cs_set_eq:NN \CJKglue \__xpinyin_save_CJKglue: }
271     \__xpinyin_restore_CJKsymbol:
272     \bool_set_false:N \l__xpinyin_enable_bool
273   }
274 }

```

```

\__xpinyin_restore_footnote: 275 \cs_new_protected:Npn \__xpinyin_restore_footnote:
276 {
277   \bool_if:NF \l__xpinyin_footnote_bool
278   { \tl_put_left:Nn \@parboxrestore { \l__xpinyin_restore_footnote_tl } }
279 }

```

```

\l_xpinyin_restore_footnote_tl 280 \tl_new:N \l_xpinyin_restore_footnote_tl
281 \tl_set:Nn \l_xpinyin_restore_footnote_tl
282 {
283   \int_compare:nNnT \tex_currentgrouptype:D = { 11 }
284     { \disablepinyin }
285 }

\l_xpinyin_CJKglue_dim 286 \dim_new:N \l_xpinyin_CJKglue_dim

\__xpinyin_single_aux:nx 287 \cs_new_protected:Npn \__xpinyin_single_aux:n #1#2
288 {
289   \__xpinyin_replace_CJKsymbol_single:n {#2}
290   #1
291   \group_end:
292 }
293 \cs_new_protected:Npn \__xpinyin_replace_CJKsymbol_single_aux:n #1
294 {
295   \bool_if:NF \l_xpinyin_enable_bool { \__xpinyin_replace_CJKsymbol: }
296   \cs_set_protected:Npn \CJKsymbol ##1
297     { \__xpinyin_single_CJKsymbol:n {##1} {#1} }
298 }
299 \cs_new_protected:Npn \__xpinyin_single_CJKsymbol:n #1#2
300 {
301   \__xpinyin_make_pinyin_box:xnn
302   { \__xpinyin_to_unicode:n {#1} } {#1} { \__xpinyin_pinyin:n {#2} }
303   \__xpinyin_save_CJKsymbol:n {#1}
304 }

\__xpinyin_replace_CJKsymbol_aux: 305 \cs_new_protected:Npn \__xpinyin_replace_CJKsymbol_aux:
306 {
307   \cs_set_eq:NN \__xpinyin_save_CJKsymbol:n \CJKsymbol
308   \cs_set_eq:NN \CJKsymbol \__xpinyin_CJKsymbol:n
309 }

\__xpinyin_restore_CJKsymbol_aux: 310 \cs_new_protected:Npn \__xpinyin_restore_CJKsymbol_aux:
311 { \cs_set_eq:NN \CJKsymbol \__xpinyin_save_CJKsymbol:n }

\__xpinyin_select_font_xetex: 312 \cs_new_protected:Npn \__xpinyin_select_font_xetex:
313 {
314   \cs_if_exist_use:cF { \l_xpinyin_coor_tl }
315   {
316     \tl_set:Nx \l_xpinyin_current_coor_tl { \l_xpinyin_coor_tl }
317     \__xpinyin_select_font_aux:
318     \int_compare:nNnF { \tex_XeTeXfonttype:D \tex_font:D } = \c_zero_int
319     {
320       \exp_last_unbraced:NNV
321       \cs_gset_eq:cN \l_xpinyin_current_coor_tl \tex_font:D
322     }
323   }
324 }

\__xpinyin_select_font_aux: 325 \cs_new_protected:Npn \__xpinyin_select_font_aux:
326 {
327   \fontsize
328   { \l_xpinyin_ratio_tl \tex_dimexpr:D \f@size pt \scan_stop: }
329   { \f@baselineskip }
330   \normalfont
331   \l_xpinyin_font_tl
332   \selectfont
333 }

\__xpinyin_to_unicode_xetex:n 334 \cs_new:Npn \__xpinyin_to_unicode_xetex:n #1
335 { \int_to_arabic:n { `#1 } }

\__xpinyin_UTF_to_unicode:n 336 \cs_new:Npn \__xpinyin_UTF_to_unicode:n #1
\__xpinyin_UTFchar_to_unicode:n 337 {
338   \int_to_arabic:n
339   { \exp_args:No \int_from_hex:n { \CJK@plane } * "100 + #1 }
340 }
341 \cs_new:Npn \__xpinyin_UTFchar_to_unicode:n #1

```

```

342 { \int_to_arabic:n { \__xpinyin_UTF_viii_to_unicode:NNNw #1 \q_stop } }
343 \cs_new:Npn \__xpinyin_UTF_viii_to_unicode:NNNw #1#2#3#4 \q_stop
344 {
345   \tl_if_empty:nTF {#4}
346     { ( `#1 - "E0 ) * "1000 + ( `#2 - "80 ) * "40 + ( `#3 - "80 ) }
347     { ( `#1 - "F0 ) * "4000 + ( `#2 - "80 ) * "1000 + ( `#3 - "80 ) * "40 + ( `#4 - "80 ) }
348 }

\__xpinyin_GBK_to_unicode:n 349 \cs_new:Npn \__xpinyin_GBK_to_unicode:n
\__xpinyin_GBKchar_to_unicode:n 350 { \CJKtu_sfd_map:nn { \CJK@plane } }
351 \cs_new:Npn \__xpinyin_GBKchar_to_unicode:n
352 { \CJKtu_char_to_unicode:n }

\__xpinyin_adjust_xeCJK_hook: 353 \cs_new_protected:Npn \__xpinyin_adjust_xeCJK_hook:
354 {
355   \cs_new_eq:NN \__xpinyin_select_font: \__xpinyin_select_font_xetex:
356   \cs_new_eq:NN \__xpinyin_to_unicode:n \__xpinyin_to_unicode_xetex:n
357   \cs_new_eq:NN \__xpinyin_char_to_unicode:n \__xpinyin_to_unicode:n
358   \cs_new_eq:NN \__xpinyin_restore_CJKsymbol: \__xpinyin_restore_CJKsymbol_aux:
359   \cs_new_eq:NN \__xpinyin_replace_CJKsymbol: \__xpinyin_replace_CJKsymbol_aux:
360   \cs_new_eq:NN \__xpinyin_replace_CJKsymbol_single:n
361     \__xpinyin_replace_CJKsymbol_single_aux:n
362   \tl_if_exist:NTF \l_xeCJK_current_font_tl
363   {
364     \tl_set:Nn \l__xpinyin_coor_tl
365     {
366       ( \tl_to_str:N \l__xpinyin_font_tl ) /
367       \l_xeCJK_current_font_tl/\l__xpinyin_ratio_tl
368     }
369   }
370   {
371     \tl_set:Nn \l__xpinyin_coor_tl
372     {
373       ( \tl_to_str:N \l__xpinyin_font_tl ) /
374       \xeCJK@family/\f@series/\f@shape/\f@size/\l__xpinyin_ratio_tl
375     }
376   }
377   \cs_new_eq:NN \__xpinyin_leavevmode: \prg_do_nothing:
378   \cs_new_protected:Npx \__xpinyin_CJKsymbol_hook:
379   {
380     \exp_not:N \makexeCJKinactive
381     \exp_not:N \__xpinyin_reselect_CJK_font:
382   }
383 }

```

`\__xpinyin_reselect_CJK_font:` `\__xpinyin_CJKsymbol_hook:` 原本无条件调用 `\xeCJK@setfont` (或 `xeCJK` 的 `\xeCJK_select_font:`) 重新选择 CJK 字体, 为测量宽度用的盒子里的汉字准备字体。问题在于, 重新选择字体的内部实现首先会清除后备字体状态, 把 `AutoFallback` 刚切换好的后备字体丢掉。于是在启用 `AutoFallback`、当前字符只有后备字体里才有字形时, 测量宽度用的盒子退回主字体, 字形缺失, 量出的是该字体 *notdef* 字形的宽度 (实测 2.8pt, 正确值为 10.0pt。这个数值与句点无关: 同一字体下任何缺少字形的字符都得到 2.8pt, 而句点的实际宽度是 2.78pt), 拼音随后被 `\box_resize_to_wd_and_ht:Nnn` 压缩到这个错误宽度。注意页面上的汉字本身不受影响: 它由盒子外的 `\__xpinyin_save_CJKsymbol:n` 输出, 此时后备字体仍然有效。

`\xeCJK_reset_fallback_font:` 是 `xeCJK` 用来标记「当前处于后备字体状态」的开关: 未启用后备字体时它等于 `\prg_do_nothing:`, `xeCJK` 切换到后备字体后它被重定义为「恢复该字体并清除标记」。据此判断: 已经处于后备字体状态时不重新选择字体, 因为当前字体正是测量宽度应当使用的字体; 否则保持原有行为。

为什么保留「否则重新选择字体」这一分支, 而不是完全删去重新选择: 实测本钩子运行时, 当前字体 (`\fontname\font`) 已经是 CJK 字体, 因为 `xeCJK` 的 `interchar` 机制 (在不同字符类别之间插入代码的机制) 在进入 CJK 类别时已经切换了字体, 而本钩子在那之后运行。已核对十余种上下文 (紧跟西文单词、标点、`\emph`、数学、`\textsf`、字号变化、

\mbox / \sbox / \hbox / \vbox, tabular, minipage, \section, pinyin scope、嵌套注音、脚注、切换 CJK 字体族), \fontname\font 无一例外都已是 CJK 字体; 删去重新选择后, 全部回归测试仍然通过。也就是说, 在当前的 xeCJK 行为下, 这一分支没有可观察的效果。保留它是为了防备 xeCJK 的内部行为发生变化 (此时 NFSS 参数确实仍是西文字体族, 一旦 xeCJK 改变切换字体的时机, 就又要重新选择字体), 而不是因为已经观察到它必不可少。相应地, 回归测试无法覆盖「这一分支被跳过」的情形。这一点记为已接受的测试覆盖缺口, 见 llmdoc/reference/build-and-test.md 的 #997 一节。

这个开关在 xeCJK 里没有独立的文档条目, 属于内部命令, 比 \xeCJK_select_font: 更容易在上游重构中改名。本包依赖的 xeCJK 内部接口清单见仓库内 llmdoc/architecture/package-architecture.md; xeCJK 一侧关于「重新选择字体会清除后备字体状态」的说明, 见仓库内 llmdoc/architecture/xecjk-architecture.md 的「后备字体 (Fallback)」一节。

```

384 \cs_new_protected:Npn \__xpinyin_reselect_CJK_font:
385 {
386   \cs_if_exist:NTF \xeCJK_reset_fallback_font:
387   {
388     \cs_if_eq:NNTF \xeCJK_reset_fallback_font: \prg_do_nothing:
389     { \__xpinyin_select_CJK_font: }
390     { }
391   }
392   { \__xpinyin_select_CJK_font: }
393 }
394 \cs_new_protected:Npn \__xpinyin_select_CJK_font:
395 {
396   \cs_if_exist_use:NF \xeCJK_select_font:
397   { \xeCJK@setfont }
398 }

\__xpinyin_adjust_CJK_hook: 399 \cs_new_protected:Npn \__xpinyin_adjust_CJK_hook:
400 {
401   \bool_if:NTF \g__xpinyin_GBK_bool
402   {
403     \cs_new_eq:NN \__xpinyin_to_unicode:n \__xpinyin_GBK_to_unicode:n
404     \cs_new_eq:NN \__xpinyin_char_to_unicode:n \__xpinyin_GBKchar_to_unicode:n
405   }
406   {
407     \cs_new_eq:NN \__xpinyin_to_unicode:n \__xpinyin_UTF_to_unicode:n
408     \cs_new_eq:NN \__xpinyin_char_to_unicode:n \__xpinyin_UTFchar_to_unicode:n
409   }
410   \cs_new_eq:NN \__xpinyin_select_font: \__xpinyin_select_font_aux:
411   \cs_new_eq:NN \__xpinyin_leavevmode: \mode_leave_vertical:
412   \cs_new_eq:NN \__xpinyin_CJKsymbol_hook: \prg_do_nothing:
413   \@ifpackageloaded { CJKpunct }
414   { \__xpinyin_adjust_CJKpunct_hook: }
415   {
416     \cs_new_eq:NN \__xpinyin_restore_CJKsymbol: \__xpinyin_restore_CJKsymbol_aux:
417     \cs_new_eq:NN \__xpinyin_replace_CJKsymbol: \__xpinyin_replace_CJKsymbol_aux:
418     \cs_new_eq:NN \__xpinyin_replace_CJKsymbol_single:n
419     \__xpinyin_replace_CJKsymbol_single_aux:n
420     \AtBeginDocument
421     {
422       \@ifpackageloaded { CJKpunct }
423       {
424         \cs_undefine:N \__xpinyin_restore_CJKsymbol:
425         \cs_undefine:N \__xpinyin_replace_CJKsymbol:
426         \cs_undefine:N \__xpinyin_replace_CJKsymbol_single:n
427         \__xpinyin_adjust_CJKpunct_hook:
428       } { }
429     }
430   }
431 }

\__xpinyin_adjust_CJKpunct_hook: 432 \cs_new_protected:Npn \__xpinyin_adjust_CJKpunct_hook:
433 {

```

```

434 \cs_new_protected:Npn \__xpinyin_restore_CJKsymbol:
435 {
436   \int_compare:nNnTF { \CJKpunct@punctstyle } = { \CJKpunct@ps@plain }
437   { \__xpinyin_restore_CJKsymbol_aux: }
438   { \cs_set_eq:NN \CJKKosymbol \__xpinyin_save_CJKsymbol:n }
439 }
440 \cs_new_protected:Npn \__xpinyin_replace_CJKsymbol:
441 {
442   \int_compare:nNnTF { \CJKpunct@punctstyle } = { \CJKpunct@ps@plain }
443   { \__xpinyin_replace_CJKsymbol_aux: }
444   {
445     \cs_set_eq:NN \__xpinyin_save_CJKsymbol:n \CJKKosymbol
446     \cs_set_eq:NN \CJKKosymbol \__xpinyin_CJKsymbol:n
447   }
448 }
449 \cs_new_protected:Npn \__xpinyin_replace_CJKsymbol_single:n ##1
450 {
451   \int_compare:nNnTF { \CJKpunct@punctstyle } = { \CJKpunct@ps@plain }
452   { \__xpinyin_replace_CJKsymbol_single_aux:n { ##1 } }
453   {
454     \bool_if:NF \l__xpinyin_enable_bool
455     { \cs_set_eq:NN \__xpinyin_save_CJKsymbol:n \CJKKosymbol }
456     \cs_set_protected:Npn \CJKKosymbol #####1
457     { \__xpinyin_single_CJKsymbol:nn { #####1 } { ##1 } }
458   }
459 }
460 }

\pinyin 461 \NewDocumentCommand \pinyin { 0 { } m }
462 {
463   \group_begin:
464   \keys_set:nn { xpinyin } {#1}
465   \l__xpinyin_font_tl
466   \l__xpinyin_format_tl { }
467   \selectfont
468   \c__xpinyin_reset_UTF_cs_tl
469   \__xpinyin_pinyin:n {#2}
470   \group_end:
471 }

```

`\__xpinyin_pinyin:n` 手册说“为了输入的方便 ü 可以用 v 代替”，那么直接写 ü 也应当能用。因此先把输入里的 ü、Û 替换成 v、V(下文称为“折叠”)，再进入原有的切分循环。

如果不这样做，`\__xpinyin_pinyin_aux:n` 不会把 ü 当作元音(判断调号位置用的 `\c__xpinyin_v_tl` 等表只含 ASCII 字母)，整个音节进入“原样输出”那个分支，声调数字不会被去掉：`\pinyin{nü3}` 排出字面的 nü3(issue 1069)。

采用折叠而不是为 ü 单独建表，是因为要同时兼顾两条引擎路线。ü 在 $\text{Xe}_{\text{L}}\text{TeX}$ 下是一个记号，在 $\text{pdf}_{\text{L}}\text{TeX}$ 加 `CJKutf8` 下是两个字节；在后者下连 `\tl_const:Nn \c__ü_tl` 这样的控制序列名都无法写出(宏包一加载就报 Invalid UTF-8 byte 一类错误)。而 `\tl_replace_all:Nnn` 在两条路线上都能把 ü 换成 v：实测 $\text{pdf}_{\text{L}}\text{TeX}$ 下 nü3 是 4 个记号，替换后同样得到 nv3。于是折叠之后，两条路线共用同一套原有逻辑，调号位置的判断和 `\__xpinyin_replace_v:n` 对 l / n 的判断都不必修改。

Ü 折叠成 V 而不是 v：`\__xpinyin_replace_v:n` 与 `\__xpinyin_num_to_tone_v:Nn` 判断前一个字母时同时接受 l、n 的大小写形式，折叠成 V 才能让 `\pinyin{Nü3}` 与 `\pinyin{Nv3}` 结果一致。

`\l__xpinyin_save_tl` 存的必须是折叠前的原始输入。“整段输入里没有声调数字”时，原样输出的就是它(`\__xpinyin_pinyin_aux:n` 的 `\quark_if_recursion_tail_stop_do:nn` 分支)。如果存的是折叠后的值，就会把用户写的 ü 换成 v：`\pinyin{nü}` 本该原样排出 nü，却排出 nv。实现折叠时曾经出现过这个回归。

```

472 \cs_new_protected:Npn \__xpinyin_pinyin:n #1

```

```

473 {
474   \__xpinyin_pinyin_init:
475   \bool_set_true:N \__xpinyin_first_bool
476   \tl_set:Nn \__xpinyin_save_tl {#1}
477   \tl_set:Nn \__xpinyin_normalized_tl {#1}
478   \tl_replace_all:Nnn \__xpinyin_normalized_tl { ü } { v }
479   \tl_replace_all:Nnn \__xpinyin_normalized_tl { Ü } { V }
480   \exp_last_unbraced:N \__xpinyin_pinyin_aux:n \__xpinyin_normalized_tl
481   \q_recursion_tail \q_recursion_stop
482 }

```

__xpinyin_normalized_tl __xpinyin_pinyin:n 折叠 ü 用的中间变量。

```

483 \tl_new:N \__xpinyin_normalized_tl

\__xpinyin_pinyin_aux:n 484 \cs_new_protected:Npn \__xpinyin_pinyin_aux:n #1
485 {
486   \quark_if_recursion_tail_stop_do:nn {#1}
487   {
488     \bool_if:NTF \__xpinyin_first_bool { \__xpinyin_save_tl }
489     { \tl_if_empty:NF \__xpinyin_item_tl { \__xpinyin_pysep_tl \__xpinyin_item_tl } }
490   }
491   \__xpinyin_if_number:nTF {#1}
492   {
493     \bool_if:NTF \__xpinyin_first_bool
494     { \bool_set_false:N \__xpinyin_first_bool }
495     { \__xpinyin_pysep_tl }
496     \__xpinyin_pre_tl
497     \__xpinyin_tone:Vn \__xpinyin_tone_tl {#1}
498     \__xpinyin_post_tl
499     \__xpinyin_pinyin_init:
500   }
501   {
502     \int_compare:nNnTF
503     { 0 \cs_if_exist_use:c { c_xpinyin_ \tl_to_str:N \__xpinyin_tone_tl _tl } } >
504     { 0 \cs_if_exist_use:c { c_xpinyin_ \tl_to_str:n {#1} _tl } }
505     { \tl_put_right:Nn \__xpinyin_post_tl {#1} }
506     {
507       \tl_set:Nn \__xpinyin_tone_tl {#1}
508       \tl_set_eq:NN \__xpinyin_pre_tl \__xpinyin_item_tl
509       \tl_clear:N \__xpinyin_post_tl
510     }
511     \tl_put_right:Nx \__xpinyin_item_tl { \__xpinyin_replace_v:n {#1} }
512   }
513   \__xpinyin_pinyin_aux:n
514 }

```

```

\__xpinyin_tone:nn 515 \cs_new_protected:Npn \__xpinyin_tone:nn #1#2
516 { \use:c { __xpinyin_num_to_tone_ #1 :Nn } {#1} {#2} }
517 \cs_generate_variant:Nn \__xpinyin_tone:nn { V }

```

```

\__xpinyin_replace_v:n 518 \cs_new:Npn \__xpinyin_replace_v:n #1
519 {
520   \str_if_eq:nnTF {#1} { v }
521   {
522     \str_case:onTF { \__xpinyin_item_tl }
523     { { l } { } { n } { } { L } { } { N } { } }
524     { \exp_not:n { ü } } { u }
525   }
526   { \exp_not:n {#1} }
527 }

```

```

\__xpinyin_pinyin_init: 528 \cs_new:Npn \__xpinyin_pinyin_init:
529 {
530   \tl_clear:N \__xpinyin_pre_tl \tl_clear:N \__xpinyin_post_tl
531   \tl_clear:N \__xpinyin_item_tl \tl_clear:N \__xpinyin_tone_tl
532 }

```

```

\__xpinyin_if_number:nTF 533 \prg_new_conditional:Npnn \__xpinyin_if_number:n #1 { TF }
534 {
535   \if_int_compare:w \c_one_int < 1 \tl_to_str:n {#1} \exp_stop_f:
536   \prg_return_true: \else: \prg_return_false: \fi:
537 }

\l__xpinyin_first_bool 538 \bool_new:N \l__xpinyin_first_bool

\c__xpinyin_a_tl 539 \tl_const:Nn \c__xpinyin_a_tl { 3 }
\c__xpinyin_o_tl 540 \tl_const:Nn \c__xpinyin_o_tl { 2 }
\c__xpinyin_e_tl 541 \tl_const:Nn \c__xpinyin_e_tl { 2 }
\c__xpinyin_i_tl 542 \tl_const:Nn \c__xpinyin_i_tl { 1 }
\c__xpinyin_u_tl 543 \tl_const:Nn \c__xpinyin_u_tl { 1 }
\c__xpinyin_v_tl 544 \tl_const:Nn \c__xpinyin_v_tl { 1 }

\__xpinyin_num_to_tone:Nn 545 \cs_new_protected:Npn \__xpinyin_num_to_tone:Nn #1#2
546 {
547   \if_case:w \int_eval:n { #2 - 1 } \exp_stop_f:
548   \= {#1} \or: \'{#1} \or: \v {#1} \or: \` {#1} \else: #1 \fi:
549 }
550 \tl_map_inline:nn { a o e u }
551 { \cs_new_eq:cN { __xpinyin_num_to_tone_ #1 :Nn } \__xpinyin_num_to_tone:Nn }
552 \cs_new:Npn \__xpinyin_num_to_tone_i:Nn #1#2
553 {
554   \if_case:w \int_eval:n { #2 - 1 } \exp_stop_f:
555   i \or: í \or: i \or: ì \else: i \fi:
556 }
557 \cs_new_protected:Npn \__xpinyin_num_to_tone_v:Nn #1#2
558 {
559   \str_case:onTF { \l__xpinyin_pre_tl }
560   { { l } { } { n } { } { L } { } { N } { } }
561   {
562     \if_case:w \int_eval:n { #2 - 1 } \exp_stop_f:
563     ü \or: ú \or: ù \or: ð \else: ü \fi:
564   }
565   { \__xpinyin_num_to_tone:Nn u {#2} }
566 }

\xpinyinsetup 567 \NewDocumentCommand \xpinyinsetup { m } { \keys_set:nn { xpinyin } {#1} }

ratio 568 \clist_map_inline:nn
vsep 569 { ratio , vsep , hsep , pysep , font , format , multiple }
hsep 570 { \keys_define:nn { xpinyin } { #1 .tl_set:c = { l__xpinyin_ #1 _tl } } }
pysep 571 \keys_define:nn { xpinyin }
font 572 { footnote .bool_set:N = \l__xpinyin_footnote_bool }
format 573 \keys_set:nn { xpinyin }
multiple 574 {
footnote 575   ratio = .4 ,
576   vsep = 1 em ,
577   pysep = \c_space_tl ,
578   font = \normalfont ,
579 }

query 是否载入查询表。这是包选项,不能在文档中途改变:数据表只在加载宏包时读一次。

580 \keys_define:nn { xpinyin }
581 { query .bool_gset:N = \g__xpinyin_query_bool }

\xpinyin_customary:nnn 582 \cs_new_protected:Npn \xpinyin_customary:nnn #1#2
\xpinyin_multiple:nnn 583 { \cs_gset_nopar:cpn { c__xpinyin_ #2 _tl } }
584 \cs_new_protected:Npn \xpinyin_multiple:nnn #1#2
585 { \cs_gset_nopar:cpn { c__xpinyin_multiple_ #2 _clist } }

586 \group_begin:
587 \cs_set_eq:NN \XPYU \xpinyin_customary:nnn
588 \cs_set_eq:NN \XPYUM \xpinyin_multiple:nnn
589 \file_input:n { xpinyin-database.def }
590 \group_end:

```

`\xpinyin_query:nn` 载入查询表时用的两个命令,与 `\xpinyin_customary:nnn` 的作用相同,只是存的是“字母加声调数字”的形式,且不再需要汉字本身作参数。

```
591 \cs_new_protected:Npn \xpinyin_query:nn #1
592 { \cs_gset_nopar:cpn { c__xpinyin_query_ #1 _tl } }
593 \cs_new_protected:Npn \xpinyin_query_multiple:nn #1
594 { \cs_gset_nopar:cpn { c__xpinyin_query_multiple_ #1 _clist } }
```

`\__xpinyin_setpinyin_strip_neutral:n` 去掉末尾表示轻声的 5,其余情形原样保留。只检查最后一个字符:出现在其他位置的 5(例如假想的 a5b)不是声调标记,不能去掉。

```
595 \cs_new_protected:Npn \__xpinyin_setpinyin_strip_neutral:n #1
596 {
597   \str_if_eq:eeTF { \str_range:nnn {#1} {-1} {-1} } { 5 }
598   { \tl_set:Nx \l__xpinyin_normalized_tl { \str_range:nnn {#1} {1} {-2} } }
599   { \tl_set:Nn \l__xpinyin_normalized_tl {#1} }
600 }
```

```
\setpinyin 601 \NewDocumentCommand \setpinyin { m m }
602 {
603   \tl_set:cn
604   { c__xpinyin_ \__xpinyin_char_to_unicode:n {#1} _tl }
605   { \__xpinyin_pinyin:n {#2} }
```

查询表也要一起修改。注音与查询用的是两张表:注音表存的是排好的注音内容,查询表存“字母加声调数字”的形式,正好就是这里第二个参数的写法。如果只改注音表,同一个“首选读音”会给出两种结果:实测 `\setpinyin{长}{chang2}` 之后 `\xpinyin` 排出 `cháng`,而 `\xpinyinvalue` 仍返回 `zhǎng`,据此算出的索引排序键也就落到了错误的分组。

判断条件用 `\g__xpinyin_query_loaded_bool`(「查询表是否已载入」),而不是 `\cs_if_exist:cT`(「这个字是否已在查询表里」)。用后者会漏掉数据库未收录的字,而这恰恰是用户需要用 `\setpinyin` 补充读音的场景:实测对数据库未收录的字(如 U+5159)用 `\setpinyin` 只更新注音表,随后 `\xpinyinvalue` 仍报 `No pinyin reading`,与上面「注音与查询给出的首选读音始终一致」的说明矛盾。所以这里要创建或覆盖对应的查询项,不能只覆盖已存在的项。

查询表未载入(没有使用 `query` 选项)时仍然不设置:这时查询命令本身就会报错,多定义一个控制序列反而会让“查询表未载入”的判断失效。

存进查询表之前,要先把读音里的 `ü` 折叠成 `v`。查询表一律用 `v` 表示 `ü` (`xpinyin.lua` 生成时就是这么写的,见该文件里的 `numbertable`),因此 `number` 形式的输出可以直接传给 `\pinyin`。若把用户写的 `ü` 原样存进去, `\setpinyin{女}{nü3}` 之后 `\xpinyinvalue{女}` 在 `number` 下返回 `nü3` 而不是 `nv3`,同一张表里就出现了两种拼写,输出也就不能保证可以直接传给 `\pinyin`。

轻声还要去掉末尾的 5。查询表里轻声是不写数字的(`xpinyin.lua` 的 `numbertable` 只产出 1–4,生成的 `xpinyin-query.def` 里没有带 5 的条目),拆分逻辑也按这个约定只把 1–4 当作声调(判断条件是 ``#1 < `5`)。若原样存入 `shi5`, `none` 形式会返回 `shi5` 而不是 `shi`,违反「不带声调」的约定。

```
606 \bool_if:NT \g__xpinyin_query_loaded_bool
607 {
608   \tl_set:Nn \l__xpinyin_normalized_tl {#2}
609   \tl_replace_all:Nnn \l__xpinyin_normalized_tl { ü } { v }
610   \tl_replace_all:Nnn \l__xpinyin_normalized_tl { Ü } { V }
611   \exp_args:Nx \__xpinyin_setpinyin_strip_neutral:n
612   { \exp_not:V \l__xpinyin_normalized_tl }
613   \cs_set_nopar:cpx { c__xpinyin_query_ \__xpinyin_char_to_unicode:n {#1} _tl }
614   { \exp_not:V \l__xpinyin_normalized_tl }
615 }
616 }
```

5.1 查询命令

下面四个命令回答“这个汉字读什么”，不排版任何内容。它们完全可展开，因此可以写在 `\csname` 与 `\endcsname` 之间、`\edef` 里或 `bib2gls` 的字段里。issue 550 提出这组命令，正是为了给中文索引计算排序键。

注意 `\index` 本身不展开参数（它先 `\@sanitize` 再原样写进 `.idx`），所以要用拼音作排序键，必须自己先用 `\edef` 展开一次；手册的“给索引条目加拼音排序键”一节给出了可用的写法。

可展开的代价是所有加工都必须在生成数据库时完成：去掉声调要把预组合字符拆开，而 `expl3` 里做这件事的 `\char_to_nfd:N` 无法在可展开的上下文里使用（实测嵌套在 `\edef` 里会直接出错）。因此查询表存的是“字母加声调数字”的形式（`zhong1.nv3`），四种输出都由它派生，运行时只做取值与字符串处理。

两个选项的当前值。`scheme` 决定声母与韵母怎么切分，`tone` 决定声调以什么形式出现。

```
\l__xpinyin_query_scheme_tl
\l__xpinyin_query_tone_tl
617 \tl_new:N \l__xpinyin_query_scheme_tl
618 \tl_new:N \l__xpinyin_query_tone_tl
619 \keys_define:nn { xpinyin }
620 {
```

分支代码里用 `#1` 而不是 `\l_keys_choice_tl`：前者在选项解析时就已替换成字面分支名，不依赖运行时的局部变量（`ctex` 的 `space` 选项曾因此出错）。

```
621   scheme .choices:nn =
622     { literal , official }
623     { \tl_set:Nn \l__xpinyin_query_scheme_tl {#1} } ,
624   tone .choices:nn =
625     { mark , number , none , v }
626     { \tl_set:Nn \l__xpinyin_query_tone_tl {#1} } ,
627 }
628 \keys_set:nn { xpinyin } { scheme = literal , tone = mark }
```

取出某个汉字的“字母加声调数字”形式。参数是汉字本身。这两个函数本身不检查查询项是否存在：由调用方 `\__xpinyin_query_first:nn` 与 `\__xpinyin_query_all:nn` 先检查，查询表未载入时由调用方报错。

```
629 \cs_new:Npn \__xpinyin_query_get:n #1
630 { \use:c { c__xpinyin_query_ \int_eval:n { `#1 } _tl } }
631 \cs_new:Npn \__xpinyin_query_get_multiple:n #1
632 { \use:c { c__xpinyin_query_multiple_ \int_eval:n { `#1 } _clist } }
```

把“字母加声调数字”拆成字母部分与声调数字。判断最后一个字符是不是数字时比较码位，而不用 `\str_if_in:nnTF`：后者不可展开，会让整个调用链失去可展开性。数字 1 到 4 的码位都小于字母 a 的码位，所以一次比较就够。轻声没有调号，这时字母部分就是整个字符串，声调部分为空。

```
633 \cs_new:Npn \__xpinyin_query_last:n #1 { \str_range:nnn {#1} {-1} {-1} }
634 \cs_new:Npn \__xpinyin_query_base:n #1
635 { \exp_args:Nf \__xpinyin_query_base:nn { \__xpinyin_query_last:n {#1} } {#1} }
636 \cs_new:Npn \__xpinyin_query_base:nn #1#2
637 {
638   \int_compare:nNnTF { `#1 } < { `5 }
639     { \str_range:nnn {#2} {1} {-2} } {#2}
640 }
641 \cs_new:Npn \__xpinyin_query_tone:n #1
642 { \exp_args:Nf \__xpinyin_query_tone:nn { \__xpinyin_query_last:n {#1} } {#1} }
643 \cs_new:Npn \__xpinyin_query_tone:nn #1#2
644 { \int_compare:nNnTF { `#1 } < { `5 } {#1} { } }
```

把“字母加声调数字”还原成带调号的形式。声调标在哪个字母上有固定规则：a 和 e 优先，其次是 ou 里的 o，否则标在最后一个元音上。这里复用包内已有的 `\__xpinyin_pinyin:n`，它正是为 `\pinyin` 处理同样的输入而写的，于是标调规则只有一份实现。注意这个分支不可展开。

```
\__xpinyin_query_mark:n
```

__xpinyin_pinyin:n 期望的是完整音节。official 模式切出来的韵母是音位形式, 可能以 v 开头(yue 还原成 ve), 而 \pinyin 把开头的 v 当成声母位置上的字母, 排出来是 uè 而不是 üè。这种组合本身不是合法音节, 所以带调号的输出对它没有意义: official 模式下的韵母改用无调形式, 并在手册里说明这一限制。

```
645 \cs_new_protected:Npn \__xpinyin_query_mark:n #1 { \__xpinyin_pinyin:n {#1} }
```

__xpinyin_query_shengmu:nn 声母与韵母的切分。两个模式的区别只在 y 和 w 算不算声母, 所以用两张表: __xpinyin_query_smlliteral... 包含 y、w, __xpinyin_query_smlofficial... 不包含。zh、ch、sh 是两个字母的声母, 必须先于单字母匹配。

用“控制序列是否存在”来查表: 两字母声母与单字母声母要分两级匹配, 用 \cs_if_exist_use:cF 逐级回退表达得最直接。(\str_case:nnF 同样可展开, 两者在这里都可用。)

```
646 \clist_map_inline:nn { zh , ch , sh }
647   { \cs_new:cpn { __xpinyin_query_sm2_ #1 : } { #1 } }
648 \clist_map_inline:nn { b,p,m,f,d,t,n,l,g,k,h,j,q,x,r,z,c,s }
649   {
650     \cs_new:cpn { __xpinyin_query_smlliteral_ #1 : } { #1 }
651     \cs_new:cpn { __xpinyin_query_smlofficial_ #1 : } { #1 }
652   }
653 \clist_map_inline:nn { y , w }
654   { \cs_new:cpn { __xpinyin_query_smlliteral_ #1 : } { #1 } }
655 \cs_new:Npn \__xpinyin_query_shengmu:nn #1#2
656   { \exp_args:Nnf \__xpinyin_query_shengmu_aux:nn {#1} { \__xpinyin_query_base:n {#2} } }
```

有几个音节没有元音, 以鼻音为核心: m、n、ng 以及 hm、hng (“嗯”“哼”一类)。它们没有声母与韵母之分, 整体作韵母处理, 声母为空。否则按字母切分会把“嗯”的 ng 切成 n 加 g, 而 g 不是韵母(实测早先的输出是 sm=[n] ym=[g4])。

这一步同时修复了另一个缺陷: 这些音节以前被切掉声母后韵母为空串, 与声调数字拼接后只剩一个单独的数字(U+5463 的韵母得到 2, tone 为 mark 时页面上排出孤立的“2”)。整体作韵母之后不再出现空韵母, 因此韵母一侧不必再加额外的判断(核对过全部 57493 个读音, 按现在的切分规则无一为空)。

```
657 \cs_new:Npn \__xpinyin_query_shengmu_aux:nn #1#2
658   {
659     \cs_if_exist:cTF { __xpinyin_query_syllabic_ #2 : }
660     { }
661     {
662       \cs_if_exist_use:cF { __xpinyin_query_sm2_ \str_range:nnn {#2} {1} {2} : }
663       {
664         \cs_if_exist_use:cF
665         { __xpinyin_query_sm1_ #1 _ \str_range:nnn {#2} {1} {1} : }
666         { }
667       }
668     }
669   }
670 \clist_map_inline:nn { m , n , ng , hm , hng }
671   { \cs_new:cpn { __xpinyin_query_syllabic_ #1 : } { #1 } }
```

韵母是去掉声母之后剩下的部分。official 模式下若声母为空, 还要把 y 或 w 的写法还原成介音。

```
672 \cs_new:Npn \__xpinyin_query_yunmu:nn #1#2
673   {
674     \exp_args:Nfff \__xpinyin_query_yunmu_aux:nnn {#1}
675     { \__xpinyin_query_shengmu:nn {#1} {#2} } { \__xpinyin_query_base:n {#2} }
676   }
677 \cs_new:Npn \__xpinyin_query_yunmu_aux:nnn #1#2#3
678   {
679     \str_if_eq:eeTF {#1} { official }
680     {
681       \tl_if_empty:nTF {#2}
682       { \__xpinyin_query_glide:n {#3} }
683       { \str_range:nnn {#3} { \tl_count:n {#2} + 1 } {-1} }
```

```

684     }
685     { \str_range:nnn {#3} { \tl_count:n {#2} + 1 } {-1} }
686 }

```

`\__xpinyin_query_glide:n` `official` 模式下把 `y` 与 `w` 的写法还原成介音,依据《汉语拼音方案》: `y` 后接 `i` 时去掉 `y`; `y` 后接 `u` 时该 `u` 实为 `ü`,按包内约定写作 `v`; `y` 后接其他元音时 `y` 变 `i`; `w` 后接 `u` 时去掉 `w`,接其他元音时 `w` 变 `u`。不以 `y` 或 `w` 开头的(`an`、`er` 一类)原样返回。

`yo` 一类按“`y` 后接其他元音”这一条处理成 `io`。这个音节的实际读法在语言学上有不同意见,本包按上面的通则统一处理,不对它做特殊处理。

```

687 \cs_new:cpn { __xpinyin_query_glide_y_i: } { i }
688 \cs_new:cpn { __xpinyin_query_glide_y_u: } { v }
689 \cs_new:cpn { __xpinyin_query_glide_w_u: } { u }
690 \cs_new:cpn { __xpinyin_query_medial_y: } { i }
691 \cs_new:cpn { __xpinyin_query_medial_w: } { u }
692 \cs_new:Npn \__xpinyin_query_glide:n #1
693 {
694     \exp_args:Nff \__xpinyin_query_glide_aux:nn
695     { \str_range:nnn {#1} {1} {1} } { \str_range:nnn {#1} {2} {-1} }
696 }
697 \cs_new:Npn \__xpinyin_query_glide_aux:nn #1#2
698 {
699     \cs_if_exist:cTF { __xpinyin_query_medial_ #1 : }
700     {
701         \cs_if_exist:cTF
702         { __xpinyin_query_glide_ #1 _ \str_range:nnn {#2} {1} {1} : }
703         {
704             \use:c { __xpinyin_query_glide_ #1 _ \str_range:nnn {#2} {1} {1} : }
705             \str_range:nnn {#2} {2} {-1}
706         }
707         { \use:c { __xpinyin_query_medial_ #1 : } #2 }
708     }
709     { #1 #2 }
710 }

```

`\__xpinyin_query_tone_apply:n` 按 `tone` 的取值决定输出形式。`number` 直接返回查询表里的原值; `none` 去掉声调数字并保留 `v`; `v` 与 `none` 相同,保留两个名字是因为“无声调”和“把 `ü` 写成 `v`”是两件事,只是在本包的存储形式下结果一致; `mark` 交给 `\__xpinyin_query_mark:n` 标调。

```

711 \cs_new:Npn \__xpinyin_query_tone_apply:n #1
712 {
713     \str_if_eq:eeTF { \l__xpinyin_query_tone_tl } { number }
714     { #1 }
715     {
716         \str_if_eq:eeTF { \l__xpinyin_query_tone_tl } { mark }
717         { \__xpinyin_query_mark_if_possible:n {#1} }
718         { \__xpinyin_query_base:n {#1} }
719     }
720 }

```

`\__xpinyin_query_mark_if_possible:n` 标调号需要有可以标调的元音字母。`m`、`n`、`ng` 一类音节(见前面的 `\__xpinyin_query_syllabic_` 表)没有元音, `\__xpinyin_pinyin:n` 找不到标调位置,会把声调数字原样排出: `\xpinyinvalue{嗯}` 得到 `ng4`,页面上会出现一个单独的 `4`。

这与声母“永远不带声调”出于同一个原因:判断依据都是“有没有可以标调的元音”。因此这里先检查,没有元音就改用无调形式(`ng` 而不是 `ng4`)。

```

721 \cs_new:Npn \__xpinyin_query_mark_if_possible:n #1
722 {
723     \__xpinyin_query_if_markable:nTF {#1}
724     { \__xpinyin_query_mark:n {#1} }
725     { \__xpinyin_query_base:n {#1} }
726 }

```

逐个字符检查有没有元音字母。这里不能用 `\tl_if_in:nnTF` 或 `\str_remove_all:nn`：两者都不可展开(实测前者在 `\edef` 里原样留下整条命令，后者把参数连花括号一起返回，长度反而变长)，而查询命令必须保持可展开。可展开的做法是用 `\str_range:nnn` 取单个字符，交给 `\str_case:enF` 比对。

```

727 \cs_new:Npn \__xpinyin_query_markable:n #1
728   { \__xpinyin_query_markable_aux:nn {#1} { 1 } }
729 \cs_new:Npn \__xpinyin_query_markable_aux:nn #1#2
730   {
731     \int_compare:nNnTF {#2} > { \str_count:n {#1} }
732     { 0 }
733     {
734       \str_case:enF { \str_range:nnn {#1} {#2} {#2} }
735       { {a}{1} {o}{1} {e}{1} {i}{1} {u}{1} {v}{1} }
736       { \exp_args:Nnf \__xpinyin_query_markable_aux:nn {#1}
737         { \int_eval:n { #2 + 1 } } }
738     }
739   }
740 \prg_new_conditional:Npnn \__xpinyin_query_if_markable:n #1 { TF }
741   {
742     \int_compare:nNnTF { \__xpinyin_query_markable:n {#1} } = { 1 }
743     { \prg_return_true: } { \prg_return_false: }
744   }

```

`\__xpinyin_query_part:nn` 按要取的部分(整个读音、首字母、声母、韵母)加工一个读音。`#1` 是部分的名字，`#2` 是查询表里的“字母加声调数字”。

首字母取的是字母部分的第一个字符，因此不受 `tone` 影响；其余三个都要经过 `\__xpinyin_query_tone_apply:n`。

```

745 \cs_new:Npn \__xpinyin_query_part:nn #1#2
746   {
747     \str_if_eq:nnTF {#1} { initial }
748     { \exp_args:Nf \__xpinyin_query_head:n { \__xpinyin_query_base:n {#2} } }
749     {

```

声母只由辅音组成，标不了调号，所以永远不带声调，四种 `tone` 取值下都一样，这个分支不经过 `\__xpinyin_query_tone_apply:n`。

这一点是实测发现的：早先把声调一律接在切出来的片段后面，结果“中”的声母排成 `zh1`——`\__xpinyin_pinyin:n` 找不到可以标调的元音，就把数字原样输出了。

```

750     \str_if_eq:nnTF {#1} { shengmu }
751     { \exp_args:Nf \__xpinyin_query_shengmu:nn \l__xpinyin_query_scheme_tl {#2} }
752     {

```

`official` 模式的韵母是音位形式，可能以 `v` 开头(`yue` 还原成 `ve`)。 `\__xpinyin_pinyin:n` 只在 `l` 与 `n` 之后才把 `v` 当作 `ü` (见 `\__xpinyin_replace_v:n`：`ju`、`qu`、`xu` 没有歧义，所以不必标 `ü`)，开头的 `v` 会排成 `u`。因此这种形式不标调，按无调形式输出，手册里说明了这一限制。判断要放在“取韵母”这个分支内部，不能移到外层，否则 `\xpinyinvalue` 也会受 `scheme` 影响，而它返回的是完整读音，与切分方式无关(实测移到外层时 `official` 下 `\xpinyinvalue` 会返回 `uang` 而不是 `wáng`)。

```

753     \str_if_eq:nnTF {#1} { yunmu }
754     { \__xpinyin_query_yunmu_out:n {#2} }
755     { \__xpinyin_query_tone_apply:n {#2} }
756   }
757 }
758 }

```

先可展开地取出要的那一段(仍是“字母加声调数字”的形式)，再统一交给 `\__xpinyin_query_tone_apply:n` 决定输出形式。这样标调这一步只在最外层出现一次：`\__xpinyin_query_mark:n` 依赖 `\__xpinyin_pinyin:n`，不可展开，若嵌在 `f` 类展开里就会出错(实测会把 `expl3` 的内部命令原样排进页面)。

```

759 \cs_new:Npn \__xpinyin_query_yunmu_join:nn #1#2 { #1 #2 }

```

韵母的输出还有一处限制。`\__xpinyin_pinyin:n`只在 `l` 与 `n` 之后把 `v` 当作 `ü` (见 `\__xpinyin_replace_v:n`), 所以韵母以 `v` 开头时标不出 `ü`: `\pinyin{v3}` 排出来是 `ǔ` 而不是 `ǖ`。“女”的韵母正是这种情况。与其不加提示地给出错误的字形, 不如输出无调形式。`official` 模式的韵母是音位形式(`yue` 还原成 `ve`), 同样不标调。

```

760 \cs_new:Npn \__xpinyin_query_yunmu_out:n #1
761 {
762   \exp_args:Nf \__xpinyin_query_yunmu_out_aux:nn
763   { \exp_args:NV \__xpinyin_query_yunmu:nn \l__xpinyin_query_scheme_tl {#1} } {#1}
764 }
765 \cs_new:Npn \__xpinyin_query_yunmu_out_aux:nn #1#2
766 {
767   \bool_lazy_or:nnTF
768   { \str_if_eq_p:ee { \l__xpinyin_query_scheme_tl } { official } }
769   { \str_if_eq_p:ee { \str_range:nnn {#1} {1} {1} } { v } }
770   { #1 }
771   {
772     \exp_args:Nf \__xpinyin_query_tone_apply:n
773     { \exp_args:Nnf \__xpinyin_query_yunmu_join:nn {#1}
774       { \__xpinyin_query_tone:n {#2} } }
775   }
776 }

```

`\__xpinyin_query_head:n` 取字母部分的首字符, 供 `\xpinyininitial` 使用。

```

777 \cs_new:Npn \__xpinyin_query_head:n #1 { \str_range:nnn {#1} {1} {1} }

```

`\xpinyinvalue`
`\xpinyininitial`
`\xpinyinshengmu`
`\xpinyinyunmu`

四个查询命令。带星号的形式返回全部读音, 用逗号分隔; 不带星号取首选读音, 与 `\xpinyin` 的取值一致。

这些命令不接受可选参数。`scheme` 与 `tone` 用 `\xpinyinsetup` 设置, 与包内其他选项一致, 并随 `TEX` 分组恢复。这样做是因为在可展开的命令里无法调用 `\keys_set:nn:keyval` 解析本身不可展开, 若把选项放进参数, 这些命令就不能再用在 `\index` 或 `bib2gls` 的字段里了。

`tone` 取 `mark` 时输出带调号, 这一步依赖 `\__xpinyin_pinyin:n`, 不可展开; 其余三种取值下四个命令都是完全可展开的。需要字面文字 (例如自己用 `\edef` 生成索引排序键) 时, 用 `number`、`none` 或 `v`。

```

778 \NewExpandableDocumentCommand \xpinyinvalue { s m }
779 { \__xpinyin_query_dispatch:nnn { value } {#1} {#2} }
780 \NewExpandableDocumentCommand \xpinyininitial { s m }
781 { \__xpinyin_query_dispatch:nnn { initial } {#1} {#2} }
782 \NewExpandableDocumentCommand \xpinyinshengmu { s m }
783 { \__xpinyin_query_dispatch:nnn { shengmu } {#1} {#2} }
784 \NewExpandableDocumentCommand \xpinyinyunmu { s m }
785 { \__xpinyin_query_dispatch:nnn { yunmu } {#1} {#2} }

```

`\__xpinyin_query_dispatch:nnn` 分派到“取首选读音”或“取全部读音”。`#2` 是星号参数。

```

786 \cs_new:Npn \__xpinyin_query_dispatch:nnn #1#2#3
787 {

```

先检查参数是不是恰好一个字符。取码位用的 ``#1` 只接受单个字符记号, 参数为空或多于一个字符时会出现 `TEX` 的底层错误 (实测 `\xpinyinvalue{中国}` 先报! `Missing \endcsnameinserted.`, 空参数报! `Improper alphabetic constant.`), 用户看不出它与本宏包有关。

这里数的是字符而不是记号: `\tl_count:n` 把 `{中}`、`\hz` 这类参数都算作一个记号, 检查会让它们通过, 随后照样出现底层错误。`\str_count:n` 先把参数转成字符串再计数, 能把这些情况也拦下来。两者都可展开。

```

788   \int_compare:nNnTF { \str_count:n {#3} } = { 1 }
789   {
790     \bool_if:nTF {#2}
791     { \__xpinyin_query_all:nn {#1} {#3} }
792     { \__xpinyin_query_first:nn {#1} {#3} }
793   }

```

```

794     { \__xpinyin_query_bad_arg:n {#3} }
795 }

```

首选读音直接查单音字表。查询表没有载入时(未使用 `query` 选项), 对应的控制序列不存在, 这里给出明确的报错而不是静默展开成空。

```

796 \cs_new:Npn \__xpinyin_query_first:nn #1#2
797 {
798     \cs_if_exist:cTF { c__xpinyin_query_ \int_eval:n { `#2 } _tl }
799     {
800         \exp_args:Nnf \__xpinyin_query_part:nn {#1}
801         { \__xpinyin_query_get:n {#2} }
802     }
803     { \__xpinyin_query_missing:n {#2} }
804 }

```

取不到值时分两种情况: 查询表根本没载入, 或者这个字不在数据库里。两者的处理办法不同, 所以报错也要分开。

报错必须走 `\msg_expandable_error:nnn` 而不是 `\msg_error:nnn`。

查询命令的用途之一是用 `\edef` 得到字面文字, 再交给别的命令 (例如作为索引排序键), 而 `\msg_error:nnn` 是 `protected` 的: 在 `\edef` 里它既不报错也不展开, 整条命令会原样留在结果里。实测未加 `query` 选项时, `\edef\x{\xpinyinvalue{汉}}` 得到的是 `\__xpinyin_query_missing:n {汉}`, 文档编译时没有任何错误, 错误的排序键却被写进了 `.idx`。

`\msg_expandable_error:nnn` 在可展开的上下文里也会发出错误, 且不下残余。

```

805 \cs_new:Npn \__xpinyin_query_bad_arg:n #1
806 { \msg_expandable_error:nnn { xpinyin } { query-bad-argument } {#1} }
807 \cs_new:Npn \__xpinyin_query_missing:n #1
808 {
809     \bool_if:NTF \g__xpinyin_query_loaded_bool
810     { \msg_expandable_error:nnn { xpinyin } { query-no-reading } {#1} }
811     { \msg_expandable_error:nn { xpinyin } { query-not-loaded } }
812 }
813 \bool_new:N \g__xpinyin_query_loaded_bool

```

星号形式把多音字的每个读音各加工一遍, 再用逗号连接。没有多音字记录的字(单音字)改用首选读音, 使星号形式对任何汉字都有合理结果。

```

814 \cs_new:Npn \__xpinyin_query_all:nn #1#2
815 {
816     \cs_if_exist:cTF { c__xpinyin_query_multiple_ \int_eval:n { `#2 } _clist }
817     {
818         \exp_args:Nnf \__xpinyin_query_all_aux:nn {#1}
819         { \__xpinyin_query_get_multiple:n {#2} }
820     }
821     { \__xpinyin_query_first:nn {#1} {#2} }
822 }

```

星号形式把每个读音各加工一遍, 再用逗号连接。

只有在结果是纯文本时才去重。tone 为 mark 的输出里含 `\__xpinyin_pinyin:n`, 它是 `protected` 的, e 类展开得不到文本(实测 `\tl_set:Nx` 之后仍是 `\__xpinyin_pinyin:n {zhong1}`), 无法作为文本放进逗号列表, 也就无法去重; 强行去重会把每一项都变成空(实测输出 `,,,`)。

因此做法是先尝试一次去重: 把加工结果放进逗号列表, 运行 `\clist_remove_duplicates:N`; 若列表变空, 说明这些项含有 `protected` 命令, 无法去重, 于是改为逐项输出。

```

823 \cs_new_protected:Npn \__xpinyin_query_all_aux:nn #1#2
824 {
825     \__xpinyin_query_collect:nn {#1} {#2}
826     \bool_lazy_and:nnTF
827     { \clist_if_empty_p:N \g__xpinyin_query_join_clist }
828     { \int_compare_p:nNn \g__xpinyin_query_count_int > { 0 } }
829     { \__xpinyin_query_all_plain:nn {#1} {#2} }

```

```

830     { \clist_use:Nn \g__xpinyin_query_join_clist { , } }
831 }

```

“先尝试去重，不行再逐项输出”比按命令与选项静态判断更可靠。会不会经过标调不只取决于查哪一部分，还取决于这个读音本身：m、n、ng 一类音节没有可以标调的元音（见 __xpinyin_query_mark_if_possible:n），韵母以 v 开头时也不标调（见 __xpinyin_query_yunmu_out_aux:nn）。这两种回退要到运行时才能确定，静态判断必然会漏掉：实测按静态判断时 \xpinyinvalue*{嗯} 在 mark 下排出 n,n,n,ng,ng,ng，而 number 下是 n2,n3,n4,ng2,ng3,ng4。

更早的判断条件是“tone 为 mark 且不是首字母”，那一版还把声母与 official 的韵母整类排除在去重之外：缺省设置下 \xpinyinshengmu*{行} 排出 h,h,h,x，而 number 下排出 h,x。声母本来不随 tone 变化，是否去重却随之改变了。

\clist_remove_duplicates:N 在项含 protected 命令时会把整个列表清空（实测两项 \prot:n {a}、\prot:n {b} 去重后项数为 0），正好可以用来判断能否去重。

不去重的分支：逐项排出，在各项之间补上逗号。

```

832 \cs_new_protected:Npn \__xpinyin_query_all_plain:nn #1#2
833 {
834     \bool_set_true:N \l__xpinyin_query_first_bool
835     \clist_map_inline:nn {#2}
836     {
837         \bool_if:NTF \l__xpinyin_query_first_bool
838         { \bool_set_false:N \l__xpinyin_query_first_bool } { , }
839         \__xpinyin_query_part:nn {#1} {##1}
840     }
841 }
842 \bool_new:N \l__xpinyin_query_first_bool

```

去重的分支。

```

843 \cs_new_protected:Npn \__xpinyin_query_collect:nn #1#2
844 {

```

收集过程要放在一个用后丢弃的盒子里。 __xpinyin_query_mark:n 经 __xpinyin_pinyin:n 直接把内容排版出来，所以这次尝试本身会产生输出。如果不放进盒子，即使随后改为逐项输出，尝试时排出的逗号也已经留在页面上了（实测输出,,, ,háng,hàng,héng,xíng）。

```

845     \hbox_set:Nn \l__xpinyin_query_probe_box
846     {
847         \clist_gclear:N \g__xpinyin_query_join_clist
848         \clist_map_inline:nn {#2}
849         {
850             \clist_gput_right:Ne \g__xpinyin_query_join_clist
851             { \__xpinyin_query_part:nn {#1} {##1} }
852         }

```

记下去重之前的项数，随后靠它区分两种“列表为空”的情形。

逗号列表会自动丢掉空项，所以全部读音的声母都为空时（“啊”“呕”一类零声母字，在 official 下还有以 y、w 开头的字），列表在去重前就已经是空的，这是正确结果。而项含有 protected 命令时，\clist_gremove_duplicates:N 会把原本非空的列表清空，这才说明无法去重。

只看去重后列表是否为空，无法区分这两种情形；实测会让 \xpinyinshengmu*{啊} 排出,,, ,只剩下分隔用的逗号。

```

853     \int_gset:Nn \g__xpinyin_query_count_int
854     { \clist_count:N \g__xpinyin_query_join_clist }
855     \clist_gremove_duplicates:N \g__xpinyin_query_join_clist
856 }
857 }
858 \int_new:N \g__xpinyin_query_count_int
859 \box_new:N \l__xpinyin_query_probe_box

```

`\g__xpinyin_query_join_clist` 去重用的逗号列表。多音字的不同读音常落在同一个首字母上 (“行” 的四个读音里三个都是 h), 逐项列出没有意义。

用全局变量: 上面的收集过程要放在盒子里 (`\__xpinyin_query_mark:n` 会直接排版), 而盒子会把局部赋值限制在盒子内部, 结果传不出来。星号形式因此不可展开——需要字面文字时用不带星号的形式。

```
860 \clist_new:N \g__xpinyin_query_join_clist

861 \cs_if_exist:NTF \ProcessKeyOptions
862   { \ProcessKeyOptions [ xpinyin ] }
863   {
864     \RequirePackage { l3keys2e }
865     \ProcessKeysOptions { xpinyin }
866   }
```

选项处理完毕后再决定要不要载入查询表。放在这里是因为 `query` 的值到这一步才确定。查询命令只能在 $\text{Xe}\text{L}\text{a}\text{T}\text{E}\text{X}$ 下使用 (本宏包支持的引擎中只有它是 Unicode 引擎), 所以先检查引擎。

```
867 \bool_if:NT \g__xpinyin_query_bool
868   {
```

判断条件写成 “必须是 $\text{Xe}\text{L}\text{a}\text{T}\text{E}\text{X}$ ” 而不是 “不能是 $\text{pdf}\text{L}\text{a}\text{T}\text{E}\text{X}$ ”: 宏包加载时的引擎检查目前只允许 $\text{Xe}\text{L}\text{a}\text{T}\text{E}\text{X}$ 与 $\text{pdf}\text{L}\text{a}\text{T}\text{E}\text{X}$, 如果日后允许别的引擎, 正向判断不会默认把它当作可用。

```
869   \sys_if_engine_xetex:TF
870   {
871     \group_begin:
872       \cs_set_eq:NN \XPYUQ \xpinyin_query:nn
873       \cs_set_eq:NN \XPYUQM \xpinyin_query_multiple:nn
874       \file_input:n { xpinyin-query.def }
875     \group_end:
876     \bool_gset_true:N \g__xpinyin_query_loaded_bool
877   }
878   { \msg_error:nn { xpinyin } { query-needs-unicode } }
879 }

880 </package>
```

第 6 节 *xpinyin.lua*

```
881 <lua>

882 xpinyin      = xpinyin or { }
883 local xpinyin = xpinyin

      计算时区2。
884 xpinyin.tzoffset = "+0000"
885 do
886   -- Compute the difference in seconds between local time and UTC.
887   local function get_timezone()
888     local now = os.time()
889     return os.difftime(now, os.time(os.date("!*t", now)))
890   end
891   -- Return a timezone string in ISO 8601:2000 standard form (+hhmm or -hhmm)
892   local function get_tzoffset(timezone)
893     local h, m = math.modf(timezone / 3600)
894     return string.format("%+.4d", 100 * h + 60 * m)
895   end
896   xpinyin.tzoffset = get_tzoffset(get_timezone())
897 end

898 xpinyin = {
899   uchar      = unicode.utf8.char,
```

²<http://lua-users.org/wiki/TimeZone>

```

900 readings = { },
901 fixreadings = {

```

为汉字“○”增加拼音。

```

902 {"U+3007", "Mandarin", "líng"},

```

下面几个字在文本里主要用作轻声的助词或后缀(“桌子”“我们”“好啊”)。v3.1 使用 Unicode 14.0 的 UniHan, 这几个字的首选读音都是轻声;之后的版本改写了 kMandarin, 给出的是单字本调(“子”为 zǐ、“啊”为 ā), 与 kHanyuPinlu 的语料统计不符(例如“子”轻声 6853 次、zǐ 1114 次)。这里保留 v3.1 的轻声读音, 避免已有文档的注音随数据更新而改变。其他字首选读音的变化多是按现行规范读音更正(如“绩”由 jī 改为 jì), 按新数据处理, 不在这里列出;需要其他读音时可用 \setpinyin。

```

903 {"U+5011", "Mandarin", "men"},
904 {"U+54C7", "Mandarin", "wa"},
905 {"U+554A", "Mandarin", "a"},
906 {"U+5566", "Mandarin", "la"},
907 {"U+5B50", "Mandarin", "zi"}
908 },

909 database = {
910   source = "https://www.unicode.org/Public/UNIDATA/UniHan.zip",
911   file    = "UniHan_Readings.txt",
912   date    = "Date: 2014-05-09 18:17:02 GMT [JHJ]",
913   version = "Unicode version: 7.0.0",
914   dbfile  = "xpinyin.db",
915   qdbfile = "xpinyin-query.db"
916 },
917 preamble = [[
918   %<<COMMENT
919   %%
920   %% Do not edit this file!
921   %% Created from UniHan database:
922   %%
923   %% $file
924   %% $date
925   %% $version
926   %%
927   %% by "texlua xpinyin.lua" on ]]
928   .. os.date("%Y-%m-%d %X ") .. xpinyin.tzoffset
929   .. "\n%%"
930 %COMMENT
931 ]}

932 local zip_open = require("zip").open

```

将 UniHan_Readings.txt³ 保存到一张表里面。文件头里的日期与 Unicode 版本会写进生成的数据库文件头。版本行在 14.0 的数据里写作 # Unicode version: 14.0.0, 在 17.0、18.0 的数据里写作 # Unicode Version 18.0.0 这种大写、无冒号的形式;只认前一种时,新数据生成的文件头会一直显示默认值 7.0.0, 因此这里两种写法都接受。

```

933 function xpinyin.maketable (txt)
934   local f = io.open(txt or xpinyin.database.file, "r")
935   if not f then
936     local source = xpinyin.database.source
937     local zfilename = source:match("[^/]+$")
938     local zfile = zip_open(zfilename)
939     if not zfile then
940       xpinyin.download(source, zfilename)
941       zfile = assert(zip_open(zfilename))
942     end
943     f = assert(zfile:open(xpinyin.database.file))
944     zfile:close()
945   end
946   local s, prop

```

³<http://http://www.unicode.org/reports/tr38/>。

```

947 for line in f:lines() do
948     s = line:explode("\t")
949     if #s == 3 then
950         prop = s[2]:sub(2)
951         if prop == "Mandarin" or
952            prop == "HanyuPinyin" or
953            prop == "XHC1983" or
954            prop == "HanyuPinlu" then
955             xpinyin.insert(s[1], prop, s[3])
956         end
957     elseif line:find("Date") then
958         xpinyin.database.date = line:match("^[%s]*(.*)")
959     elseif line:find("^%s*Unicode [Vv]ersion") then
960         xpinyin.database.version = line:match("^[%s]*(.*)")
961     end
962 end
963 f:close()
964 if xpinyin.fixreadings then
965     for _, s in pairs(xpinyin.fixreadings) do
966         xpinyin.insert(s[1], s[2], s[3])
967     end
968 end
969 end

```

下载 Unihan.zip。

```

970 function xpinyin.download (source, zip)
971     print("\nRetrieving Unihan Database from\n", source)
972     local ok, http = pcall(require, "socket.http")
973     if ok then
974         local lok, ltn12 = pcall(require, "ltn12")
975         if lok then
976             local fh = io.open(zip, "wb")
977             if fh then
978                 local status = http.request{
979                     url = source,
980                     sink = ltn12.sink.file(fh)
981                 }
982                 if status then return end
983             end
984         end
985     end
986     local ret = os.execute('curl -fsSL -o "' .. zip .. '" "' .. source .. '"')
987     if ret == 0 or ret == true then return end
988     ret = os.execute('wget -q -O "' .. zip .. '" "' .. source .. '"')
989     if ret == 0 or ret == true then return end
990     error(
991         "Failed to download '" .. zip .. "'. Try one of:\n"
992         .. ' ' - Run: texlua --socket xpinyin.lua\n'
993         .. ' ' - Install curl or wget\n'
994         .. ' ' - Manually place "' .. zip .. '" in the current directory"
995     )
996 end

```

往拼音表中加入项目。

```

997 function xpinyin.insert (unicode, prop, value)
998     local index = tonumber(unicode:match("%x+$"), 16)
999     if not xpinyin.readings[index] then
1000         xpinyin.readings[index] = { }
1001     end
1002     xpinyin.readings[index][prop] = value
1003 end

```

把带调号的拼音转成“字母加声调数字”的形式,例如 zhōng 转成 zhong1、nǚ 转成 nv3。这一步在生成数据库时做,查询命令运行时只需查表,于是可以保持完全可展开(见查询命令一节的说明)。

Unihan 里的拼音用的是预组合字符, 例如 $\bar{o}$ 是单独一个码位 U+014D, 而不是 o 加上长音符两个码位。因此这里用一张显式的对照表, 而不依赖 Unicode 规范化: texlua 没有提供规范化函数。表中列出 29 个带附加符号的字符, 每项给出“去掉附加符号后的字母”和“声调数字”。其中 28 个是当前数据库实际用到的; $\bar{u}$ (U+01D6) 在当前数据里没有出现, 为把 $\bar{u}$ 的四声补齐而一并列出。

$\bar{u}$ 及其带调字符需要单独说明。按包内 \pinyin 的输入约定, $\bar{u}$ 写作 v (手册里的 \pinyin{nv3hai2zi}), 所以 $\bar{u}$ 的字母部分是 v 而不是 u: 若写成 u, nǚ 会变成 nu3, 传给 \pinyin 得到的是“努”而不是“女”。

轻声没有调号, 声调数字为空字符串, 例如 de 仍然是 de。

```
1004 xpinyin.numbertable = {
1005   ["ā"] = {"a", "1"}, ["á"] = {"a", "2"},
1006   ["ǎ"] = {"a", "3"}, ["à"] = {"a", "4"},
1007   ["ē"] = {"e", "1"}, ["é"] = {"e", "2"},
1008   ["ě"] = {"e", "3"}, ["è"] = {"e", "4"},
1009   ["ī"] = {"i", "1"}, ["í"] = {"i", "2"},
1010   ["ǐ"] = {"i", "3"}, ["ì"] = {"i", "4"},
1011   ["ō"] = {"o", "1"}, ["ó"] = {"o", "2"},
1012   ["ǒ"] = {"o", "3"}, ["ò"] = {"o", "4"},
1013   ["ū"] = {"u", "1"}, ["ú"] = {"u", "2"},
1014   ["ǔ"] = {"u", "3"}, ["ù"] = {"u", "4"},
1015   ["ǖ"] = {"v", "1"}, ["ǘ"] = {"v", "2"},
1016   ["ǚ"] = {"v", "3"}, ["ǜ"] = {"v", "4"},
1017   ["ü"] = {"v", "" },
```

ń, ǎ, ñ 和 ń 用在“嗯”一类字上, 声调直接标在鼻音字母上。

```
1018   ["ń"] = {"n", "2"}, ["ǎ"] = {"n", "3"}, ["ñ"] = {"n", "4"},
1019   ["m"] = {"m", "2"}
1020 }
```

少数组合并没有对应的预组合字符, 只能以“字母加组合符”两个码位出现。本数据库里的实例是 U+5463 的读音 ñ: Unicode 有 ń(U+1E3F), 却没有 m 加钝音符的预组合形式。因此除上表之外, 还要识别作为独立码位出现的组合符。

```
1021 xpinyin.combining = {
1022   ["\u{0304}"] = "1", ["\u{0301}"] = "2",
1023   ["\u{030C}"] = "3", ["\u{0300}"] = "4"
1024 }
1025 xpinyin.diaeresis = "\u{0308}"
1026 function xpinyin.numbered (pinyin)
1027   local base, tone = { }, ""
1028   for _, ch in ipairs(xpinyin.chars(pinyin)) do
1029     local item = xpinyin.numbertable[ch]
1030     if item then
1031       base[#base + 1] = item[1]
1032       if item[2] ~= "" then tone = item[2] end
1033     elseif xpinyin.combining[ch] then
1034       tone = xpinyin.combining[ch]
1035     elseif ch == xpinyin.diaeresis then
```

作为独立码位出现的分音符只会跟在 u 后面, 同样按 \pinyin 的约定折叠成 v。

```
1036       if base[#base] == "u" then base[#base] = "v" end
1037     else
1038       base[#base + 1] = ch
1039     end
1040   end
1041   return table.concat(base) .. tone
1042 end
```

把字符串按码位切成单个字符。

```
1043 function xpinyin.chars (s)
1044   local t = { }
1045   for _, cp in utf8.codes(s) do t[#t + 1] = xpinyin.uchar(cp) end
1046   return t
1047 end
```

输出需要的格式文件。

```

1048 function xpinyin.output (db)
1049   local f = assert(io.open(db or xpinyin.database.dbfile, "w"))
1050   local preamble = xpinyin.preamble:gsub("%$(%w+)", xpinyin.database)
1051   f:write(preamble, "\n")
1052   local hanzi, pinyin
1053   local mt, qt = { }, { }
1054   for index, pyt in xpinyin.pairsByKeys(xpinyin.readings) do
1055     pinyin = assert(xpinyin.grep(pyt))
1056     hanzi = xpinyin.uchar(index)
1057     f:write("\\XPYU{" .. hanzi, "}" .. index, "}" .. pinyin, "\n")

```

查询用的“字母加声调数字”形式写进另一个文件。查询表只有用到查询命令的文档才需要，分开存放使 query 选项可以按需载入，不增加现有用户的载入开销。

```

1058   qt[#qt + 1] = "\\XPYUQ{" .. index .. "}"
1059   .. xpinyin.numbered(pinyin) .. "}"
1060   pinyin = xpinyin.multiple(pyt)
1061   if pinyin then
1062     mt[#mt + 1] = "\\XPYUM{" .. hanzi .. "}" .. index .. "}" .. pinyin .. "}"
1063     local nums = { }
1064     for _, py in ipairs(pinyin:explode(", ")) do
1065       nums[#nums + 1] = xpinyin.numbered(py)
1066     end
1067     qt[#qt + 1] = "\\XPYUQM{" .. index .. "}"
1068     .. table.concat(nums, ", ") .. "}"
1069   end
1070 end
1071 f:write(table.concat(mt, "\n"), "\n")
1072 f:close()
1073 local q = assert(io.open(xpinyin.database.qdbfile, "w"))
1074 q:write(preamble, "\n")
1075 q:write(table.concat(qt, "\n"), "\n")
1076 q:close()
1077 end

```

将表按照索引排序，代码来源于 *Programming in Lua*。

```

1078 function xpinyin.pairsByKeys (t, f)
1079   local a = { }
1080   for n in pairs(t) do a[#a + 1] = n end
1081   table.sort(a, f)
1082   local i = 0 -- iterator variable
1083   return function () -- iterator function
1084     i = i + 1
1085     return a[i], t[a[i]]
1086   end
1087 end

```

按照 Mandarin、XHC1983、HanyuPinyin 的顺序选择最常用的拼音。HanyuPinlu 的质量较差，不采用。

```

1088 function xpinyin.grep (pyt)
1089   if pyt.Mandarin then
1090     return pyt.Mandarin:match("%S+"), "Mandarin"
1091   elseif pyt.XHC1983 then
1092     return pyt.XHC1983:match(":(%S+)"), "XHC1983"
1093   elseif pyt.HanyuPinyin then
1094     return pyt.HanyuPinyin:match(":(^[^,%S]+)"), "HanyuPinyin"
1095   end
1096 end

```

根据 XHC1983 和 HanyuPinyin 选出多音字。

```

1097 function xpinyin.multiple (pyt)
1098   if pyt.XHC1983 then
1099     local s = pyt.XHC1983:explode()
1100     if s[2] then
1101       local t = { }
1102       for i, v in ipairs(s) do
1103         t[#t + 1] = v:explode(":")[2]

```

```

1104     end
1105     return xpinyin.unique(t), "XHC1983"
1106 end
1107 elseif pyt.HanyuPinyin and pyt.HanyuPinyin:find("%D,") then
1108     local t = { }
1109     for _, v in ipairs(pyt.HanyuPinyin:explode()) do
1110         for _, py in ipairs(v:explode(":")[2]:explode(",")) do
1111             t[#t + 1] = py
1112         end
1113     end
1114     return xpinyin.unique(t), "HanyuPinyin"
1115 end
1116 end

```

删除掉数组中的重复元素。

```

1117 function xpinyin.unique (t)
1118     local rt = xpinyin.remove_duplicate(t)
1119     if #rt > 1 then
1120         return table.concat(rt, ",")
1121     end
1122 end
1123 function xpinyin.remove_duplicate (t)
1124     local ht = { }
1125     local nt = { }
1126     for i, v in ipairs(t) do
1127         if not ht[v] then
1128             nt[#nt + 1] = v
1129             ht[v] = true
1130         end
1131     end
1132     return nt
1133 end
1134 xpinyin.maketable()
1135 xpinyin.output()
1136 </lua>

```

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	
\"	64, 65, 66, 67, 68
\'	548
\\	5, 6, 7, 20, 21, 22, 28, 33, 1057, 1058, 1062, 1067
\`	548
A	
\AtBeginDocument	420
\AtEndOfPackage	124, 129, 136, 149, 154
B	
bool commands:	
\bool_gset_true:N	143, 876
\bool_if:NTF	221, 234, 248, 250, 265, 267, 277, 295, 401, 454, 488, 493, 606, 809, 837, 867
\bool_if:nTF	790
\bool_lazy_and:nnTF	826
\bool_lazy_or:nnTF	14, 767
\bool_new:N	122, 241, 242, 538, 813, 842
\bool_set_false:N	266, 272, 494, 838
\bool_set_true:N	243, 249, 260, 475, 834
box commands:	
\box_ht:N	188
\box_move_up:nn	190
\box_new:N	159, 160, 859
\box_resize_to_wd_and_ht:Nnn	12, 186
\box_use_drop:N	193
\box_wd:N	164, 183, 184, 187, 192
C	
char commands:	
\char_set_catcode:nn	98
\char_set_catcode_active:n	99, 109
\char_set_lccode:nn	113
\char_to_nfd:N	18
\char_value_catcode:n	98
\CJKglue	235, 254, 255, 257, 269, 270
\CJKsymbol	438, 445, 446, 455, 456
\CJKsymbol	296, 307, 308, 311
CJKtu commands:	
\CJKtu_char_to_unicode:n	352
\CJKtu_sfd_map:nn	350
clist commands:	
\clist_count:N	854
\clist_gclear:N	847
\clist_gput_right:Nn	850
\clist_gremove_duplicates:N	24, 855
\clist_if_empty_p:N	827
\clist_if_exist:NTF	178
\clist_map_inline:nn	568, 646, 648, 653, 670, 835, 848
\clist_new:N	860
\clist_remove_duplicates:N	23, 24
\clist_use:Nn	830
cs commands:	
\cs_generate_variant:Nn	203, 211, 517
\cs_gset_eq:NN	321
\cs_gset_nopar:Npn	583, 585, 592, 594
\cs_if_eq:NNTF	269, 388
\cs_if_exist:NTF	17, 36, 72, 84, 386, 659, 699, 701, 798, 816, 861
\cs_if_exist_use:N	503, 504
\cs_if_exist_use:NTF	19, 314, 396, 662, 664
\cs_new:Npn	334, 336, 341, 343, 349, 351, 518, 528, 552, 629, 631, 633, 634, 636, 641, 643, 647, 650, 651, 654, 655, 657, 671, 672, 677, 687, 688, 689, 690, 691, 692, 697, 711, 721, 727, 729, 745, 759, 760, 765, 777, 786, 796, 805, 807, 814
\cs_new_eq:NN	355, 356, 357, 358, 359, 360, 377, 403, 404, 407, 408, 410, 411, 412, 416, 417, 418, 551
\cs_new_protected:Npn	70, 75, 80, 82, 102, 104, 110, 161, 166, 204, 206, 244, 275, 287, 293, 299, 305, 310, 312, 325, 353, 384, 394, 399, 432, 434, 440, 449, 472, 484, 515, 545, 557, 582, 584, 591, 593, 595, 645, 823, 832, 843
\cs_new_protected:Npx	378
\cs_set_eq:NN	117, 201, 254, 255, 270, 307, 308, 311, 438, 445, 446, 455, 587, 588, 872, 873
\cs_set_nopar:Npx	613
\cs_set_protected:Npn	296, 456
\cs_undefine:N	424, 425, 426
D	
dim commands:	
\dim_compare:nNnTF	182
\dim_new:N	286
\disablepinyin	3, 263, 284
E	
else commands:	
\else:	536, 548, 555, 563
\enablepinyin	3, 215, 227, 246
exp commands:	
\exp_args:Nc	96
\exp_args:Nf	81, 86, 635, 642, 748, 762, 772
\exp_args:Nff	694
\exp_args:Nnf	656, 736, 773, 800, 818
\exp_args:Nnff	674
\exp_args:No	339
\exp_args:NV	751, 763
\exp_args:Nx	78, 611
\exp_last_unbraced:NNV	320
\exp_last_unbraced:NV	480
\exp_not:N	380, 381

- `\exp_not:n` 524, 526, 612, 614
`\exp_stop_f:` 535, 547, 554, 562
- ### F
- fi commands:
`\fi:` 536, 548, 555, 563
 file commands:
`\file_input:n` 589, 874
`font` 3, 568
`\fontsize` 327
`footnote` 3, 568
`format` 3, 568
- ### G
- group commands:
`\group_begin:` 108, 112, 225, 232, 463, 586, 871
`\group_end:` 116, 120, 229, 291, 470, 590, 875
- ### H
- hbox commands:
`\hbox_overlap_right:n` 169
`\hbox_set:Nn` 163, 171, 173, 845
`\hbox_to_wd:nn` 192
`hsep` 3, 568
- ### I
- `\i` 60, 61, 62, 63
 if commands:
`\if_case:w` 547, 554, 562
`\if_int_compare:w` 535
`\IfBooleanTF` 223, 239
 int commands:
`\int_case:nn` 88
`\int_compare:nNnTF`
 283, 318, 436, 442, 451, 502, 638, 644, 731, 742, 788
`\int_compare_p:nNn` 828
`\int_eval:n` 81, 547, 554, 562, 630, 632, 737, 798, 816
`\int_from_hex:n` 339
`\int_gset:Nn` 853
`\int_new:N` 858
`\int_to_arabic:n` 335, 338, 342
`\c_one_int` 535
`\c_zero_int` 318
- ### IOW commands:
- `\iow_char:N` 22
- ### K
- keys commands:
`\l_keys_choice_tl` 18
`\keys_define:nn` 570, 571, 580, 619
`\keys_set:nn` 22, 214, 226, 233, 464, 567, 573, 628
- ### M
- `\makexeCJKinactive` 380
 mode commands:
`\mode_leave_vertical:` 220, 411
 msg commands:
`\msg_critical:nn` 17
- `\msg_error:nn` 11, 878
`\msg_error:nnn` 23
`\msg_expandable_error:nn` 811
`\msg_expandable_error:nnn` 23, 806, 810
`\msg_new:nnn` 3, 12, 18, 24, 26, 31
`multiple` 3, 568
- ### N
- `\n` 929, 971, 991, 992, 993, 1051, 1057, 1071, 1074, 1075
`\NewDocumentCommand` 36, 218, 246, 263, 461, 567, 601
`\NewDocumentEnvironment` 212
`\NewExpandableDocumentCommand` 778, 780, 782, 784
`\normalfont` 330, 578
`\nRetrieving` 971
- ### O
- or commands:
`\or:` 548, 555, 563
- ### P
- `\pinyin` 2, 461
`pinyinscope` 2, 212
 prg commands:
`\prg_do_nothing:` 12, 377, 388, 412
`\prg_new_conditional:Npnn` 533, 740
`\prg_return_false:` 536, 743
`\prg_return_true:` 536, 743
`\ProcessKeyOptions` 861, 862
`\ProcessKeysOptions` 865
 prop commands:
`\prop_const_from_keyval:Nn` 38
`\prop_map_function:NN` 128, 135, 153
`pysep` 3, 568
- ### Q
- quark commands:
`\quark_if_recursion_tail_stop_do:nn` 14, 486
`\q_recursion_stop` 481
`\q_recursion_tail` 481
`\q_stop` 342, 343
`query` 5, 580
- ### R
- `ratio` 3, 568
`\RequirePackage` 37, 134, 148, 152, 864
- ### S
- scan commands:
`\scan_stop:` 328
`scheme` 5
`\selectfont` 332, 467
`\setpinyin` 2, 601
 skip commands:
`\skip_horizontal:n` 245
 str commands:
`\str_case:nnTF` 19, 21, 522, 559, 734
`\str_count:n` 22, 731, 788
`\str_if_eq:nnTF` ... 520, 597, 679, 713, 716, 747, 750, 753

\str_if_eq_p:nn	768, 769	\xeCJK@setfont	12, 397
\str_if_in:nnTF	18	\xpinyin	2, 3, 6, 17, 22
\str_range:nnn	21, 597, 598, 633, 639, 662, 665, 683, 685, 695, 702, 704, 705, 734, 769, 777	\xpinyin*	2
\str_remove_all:nn	21	\xpinyininitial	4, 22
sys commands:		\xpinyininitial*	4
\c_sys_engine_str	13	\xpinyinsetup	2, 5, 22
\sys_if_engine_pdftex:TF	198	\xpinyinshengmu	4, 5
\sys_if_engine_pdftex_p:	16	\xpinyinvalue	4, 6, 17, 21
\sys_if_engine_xetex:TF	146, 869	\xpinyinvalue*	4
\sys_if_engine_xetex_p:	15	\xpinyinyunmu	4, 5
T			
\t	948	tex commands:	
TeX and L ^A T _E X _{2ϵ} commands:		\tex_currentgrouptype:D	283
\@ifpackagelater	10	\tex_dimexpr:D	328
\@ifpackageloaded	123, 126, 132, 413, 422	\tex_font:D	318, 321
\@parboxrestore	278	\tex_hss:D	193
\@sanitize	5, 18	\tex_lowercase:D	114
\@tabacckludge	7, 40, 41, 43, 44, 45, 47, 48, 49, 51, 52, 53, 55, 56, 57, 59, 60, 61, 63, 65, 66, 68	\tex_undefined:D	201
\@tempa	6	\tex_XeTeXfonttype:D	318
\a	7	tl commands:	
\begin	2	\c_space_tl	577
\CJK@plane	7, 201, 339, 350	\tl_clear:N	509, 530, 531
\CJKglue	3	\tl_const:Nn	73, 539, 540, 541, 542, 543, 544
\CJKpunct@ps@plain	436, 442, 451	\tl_count:n	22, 88, 683, 685
\CJKpunct@punctstyle	436, 442, 451	\tl_gput_right:Nn	97, 117
\csname	18	\tl_head:n	78
\disablepinyin	3	\tl_if_empty:NTF	252, 489
\edef	18, 21–23	\tl_if_empty:nTF	345, 681
\emph	12	\tl_if_exist:NTF	362
\enablepinyin	3	\tl_if_in:nnTF	21
\end	2	\tl_map_inline:nn	550
\endcsname	18	\tl_new:N	107, 121, 197, 280, 483, 617, 618
\f@baselineskip	329	\tl_put_left:Nn	278
\f@series	374	\tl_put_right:Nn	138, 200, 505, 511
\f@shape	374	\tl_replace_all:Nnn	14, 478, 479, 609, 610
\f@size	328, 374	\tl_set:Nn	23, 281, 316, 364, 371, 476, 477, 507, 598, 599, 603, 608, 623, 626
\font	12, 13	\tl_set_eq:NN	508
\fontname	12, 13	\tl_to_str:N	366, 373, 503
\footnote	3	\tl_to_str:n	105, 504, 535
\hbox	13	\tl_use:N	141
\index	5, 18, 22	tone	4
\makeatletter	6	U	
\makeatother	6	\u	1022, 1023, 1025
\mbox	13	use commands:	
\normalfont	1, 3	\use:N	105, 208, 516, 630, 632, 704, 707
\pinyin	2–5, 17–19, 28	\use_i:nn	239
\sbox	13	V	
\section	13	\v	42, 46, 50, 54, 58, 62, 67, 548
\setpinyin	2, 6, 17, 26	vsep	3, 568
\textsfs	12	X	
\usepackage	5	xeCJK commands:	
\vbox	13	\l_xeCJK_current_font_tl	362, 367
\xeCJK@family	374	\xeCJK_reset_fallback_font:	386, 388
		\xeCJK_select_font:	396

\xpinyin 2, 218
 xpinyin commands:
 \xpinyin_customary:nnn 17, 582, 582, 587
 \xpinyin_multiple:nnn 582, 584, 588
 \xpinyin_query:nn 591, 591, 872
 \xpinyin_query_multiple:nn 591, 593, 873
 xpinyin internal commands:
 \c_xpinyin_a_tl 539
 _xpinyin_adjust_CJK_hook: ... 129, 140, 154, 399, 399
 _xpinyin_adjust_CJKpunct_hook: .. 414, 427, 432, 432
 _xpinyin_adjust_xeCJK_hook: 124, 149, 353, 353
 _xpinyin_char_to_unicode:n .. 357, 404, 408, 604, 613
 _xpinyin_CJKglue: 244, 244, 255, 269
 \l_xpinyin_CJKglue_dim 184, 187, 235, 257, 286
 _xpinyin_CJKsymbol:n 204, 204, 308, 446
 _xpinyin_CJKsymbol:nn 205, 206, 211
 _xpinyin_CJKsymbol_hook: 12, 172, 378, 412
 \l_xpinyin_coor_tl 314, 316, 364, 371
 \l_xpinyin_current_coor_tl 316, 321
 \c_xpinyin_e_tl 539
 \l_xpinyin_enable_all_bool ... 221, 241, 248, 249, 266
 \l_xpinyin_enable_bool
 234, 241, 250, 260, 267, 272, 295, 454
 \l_xpinyin_first_bool 475, 488, 493, 494, 538
 \l_xpinyin_font_tl 331, 366, 373, 465
 \l_xpinyin_footnote_bool 277, 572
 \l_xpinyin_format_tl 177, 466
 \g_xpinyin_GBK_bool 122, 143, 401
 _xpinyin_GBK_char:nn 75, 75, 135
 _xpinyin_GBK_char_aux:nn 78, 80
 _xpinyin_GBK_char_aux:nnn 81, 82
 _xpinyin_GBK_char_def:nnn 86, 102
 _xpinyin_GBK_to_unicode:n 349, 349, 403
 _xpinyin_GBKchar_to_unicode:n 349, 351, 404
 \l_xpinyin_hsep_tl 245, 252
 \c_xpinyin_i_tl 539
 _xpinyin_if_number:n 533
 _xpinyin_if_number:nTF 491, 533
 \l_xpinyin_item_tl 489, 508, 511, 522, 531
 _xpinyin_leavevmode: 168, 377, 411
 _xpinyin_make_pinyin_box:nnn . 166, 166, 203, 208, 301
 \l_xpinyin_multiple_tl 179
 \l_xpinyin_normalized_tl
 . 477, 478, 479, 480, 483, 598, 599, 608, 609, 610, 612, 614
 _xpinyin_num_to_tone:Nn 545, 545, 551, 565
 _xpinyin_num_to_tone_i:Nn 552
 _xpinyin_num_to_tone_v:Nn 14, 557
 \c_xpinyin_o_tl 539
 _xpinyin_pinyin:n 15, 18-24, 302, 469, 472, 472, 605, 645
 _xpinyin_pinyin_aux:n 14, 480, 484, 484, 513
 \l_xpinyin_pinyin_box_hook_tl 138, 175, 197, 200
 _xpinyin_pinyin_init: 474, 499, 528, 528
 \l_xpinyin_post_tl 498, 505, 509, 530
 \l_xpinyin_pre_tl 496, 508, 530, 559
 \l_xpinyin_pysep_tl 489, 495
 _xpinyin_query_all:nn 18, 791, 814
 _xpinyin_query_all_aux:nn 818, 823
 _xpinyin_query_all_plain:nn 829, 832
 _xpinyin_query_bad_arg:n 794, 805
 _xpinyin_query_base:n 633, 634, 656, 675, 718, 725, 748
 _xpinyin_query_base:nn 633, 635, 636
 \g_xpinyin_query_bool 581, 867
 _xpinyin_query_collect:nn 825, 843
 \g_xpinyin_query_count_int 828, 853, 858
 _xpinyin_query_dispatch:nnn
 779, 781, 783, 785, 786, 786
 _xpinyin_query_first:nn 18, 792, 796, 821
 \l_xpinyin_query_first_bool 834, 837, 838, 842
 _xpinyin_query_get:n 629, 629, 801
 _xpinyin_query_get_multiple:n 629, 631, 819
 _xpinyin_query_glide:n 682, 687, 692
 _xpinyin_query_glide_aux:nn 694, 697
 _xpinyin_query_head:n 748, 777, 777
 _xpinyin_query_if_markable:n 740
 _xpinyin_query_if_markable:nTF 721, 723
 \g_xpinyin_query_join_clist
 827, 830, 847, 850, 854, 855, 860
 _xpinyin_query_last:n 633, 633, 635, 642
 \g_xpinyin_query_loaded_bool . 17, 606, 809, 813, 876
 _xpinyin_query_mark:n ... 20, 21, 24, 25, 645, 645, 724
 _xpinyin_query_mark_if_possible:n 24, 717, 721, 721
 _xpinyin_query_markable:n 721, 727, 742
 _xpinyin_query_markable_aux:nn 728, 729, 736
 _xpinyin_query_missing:n 803, 807
 _xpinyin_query_part:nn 745, 745, 800, 839, 851
 \l_xpinyin_query_probe_box 845, 859
 \l_xpinyin_query_scheme_tl 617, 751, 763, 768
 _xpinyin_query_shengmu:nn 646, 655, 675, 751
 _xpinyin_query_shengmu_aux:nn 656, 657
 _xpinyin_query_smlliteral... 19
 _xpinyin_query_smlloficial... 19
 _xpinyin_query_syllabic_ 20
 _xpinyin_query_tone:n 633, 641, 774
 _xpinyin_query_tone:nn 633, 642, 643
 _xpinyin_query_tone_apply:n . 21, 711, 711, 755, 772
 \l_xpinyin_query_tone_tl 617, 713, 716
 _xpinyin_query_yunmu:nn 646, 672, 763
 _xpinyin_query_yunmu_aux:nnn 674, 677
 _xpinyin_query_yunmu_join:nn 759, 773
 _xpinyin_query_yunmu_out:n 754, 760
 _xpinyin_query_yunmu_out_aux:nn 24, 762, 765
 \l_xpinyin_ratio_tl 328, 367, 374
 _xpinyin_replace_CJKsymbol:
 258, 295, 359, 417, 425, 440
 _xpinyin_replace_CJKsymbol_aux:
 305, 305, 359, 417, 443
 _xpinyin_replace_CJKsymbol_single:n
 289, 360, 418, 426, 449
 _xpinyin_replace_CJKsymbol_single_aux:n
 293, 361, 419, 452
 _xpinyin_replace_v:n 14, 21, 22, 511, 518, 518
 _xpinyin_reselect_CJK_font: 381, 384, 384

\c_xpinyin_reset_UTF_catcode_tl	97, 107, 141	__xpinyin_to_unicode_xetex:n	334, 334, 356
\c_xpinyin_reset_UTF_cs_tl	117, 121, 139, 468	__xpinyin_tone:nn	497, 515, 515, 517
__xpinyin_restore_CJKsymbol: .	271, 358, 416, 424, 434	\c_xpinyin_tone_prop	38, 128, 135, 153
__xpinyin_restore_CJKsymbol_aux:		\l_xpinyin_tone_tl	497, 503, 507, 531
.....	310, 310, 358, 416, 437	\c_xpinyin_u_tl	539
__xpinyin_restore_footnote:	259, 275, 275	__xpinyin_UTF_char:nn	70, 70, 77, 128, 153
\l_xpinyin_restore_footnote_tl	278, 280	__xpinyin_UTF_to_unicode:n	336, 336, 407
__xpinyin_save_CJKglue:	254, 270	__xpinyin_UTF_viii_to_unicode:NNnw	342, 343
__xpinyin_save_CJKsymbol:n		__xpinyin_UTFchar_to_unicode:n	336, 341, 408
.....	12, 172, 209, 303, 307, 311, 438, 445, 455	\c_xpinyin_v_tl	14, 539
\l_xpinyin_save_tl	14, 476, 488	\l_xpinyin_vsep_tl	190
__xpinyin_save_UTF_cs:Nn	96, 108, 110	__xpinyin_width:Nn	161, 161, 235, 257
__xpinyin_select_CJK_font:	389, 392, 394	\xpinyininitial	4, 778
__xpinyin_select_font:	176, 355, 410	\xpinyinsetup	2, 567
__xpinyin_select_font_aux:	317, 325, 325, 410	\xpinyinshengmu	4, 778
__xpinyin_select_font_xetex:	312, 312, 355	\xpinyinvalue	4, 778
__xpinyin_setpinyin_strip_neutral:n .	595, 595, 611	\xpinyinyunmu	4, 778
__xpinyin_single_aux:nn	236, 287, 287	\XPYU	587
__xpinyin_single_CJKsymbol:nn	297, 299, 457	\XPYUM	588
\l_xpinyin_tmpa_box ..	159, 163, 164, 171, 184, 187, 192	\XPYUQ	872
\l_xpinyin_tmpb_box	159, 173, 183, 186, 188, 193	\XPYUQM	873
__xpinyin_to_unicode:n ..	7, 205, 302, 356, 357, 403, 407		