# **输入法使用说明**

索引

1. 简介
2. 主要特色
3. 安装
4. 输入法调用热键
5. 深山红叶五笔特殊符号输入方法
6. 当前日期、时间输入方法
7. 符号通用快捷输入
8. Z键扩展简码
9. 容错码
10. 编辑词库
11. 编辑扩展词库
12. 添加用户自定义短语
13. 小狼毫常用参数修改指南

## 简介

这是一套按照绝大多数人习惯配置优化好的小狼毫。

小狼毫是一款优秀的开源免费输入法平台，可定制性极强，无任何广告骚扰和隐私泄露的风险，但要配置好它，对于不少人是比较困难的。为此，我们把它进行了全方位配置，以做到各种需求都可拿来即用，不需折腾。

对于喜欢折腾的朋友，本套小狼毫的所有方案和码表都是开放的，同时附带了丰富的功能导向的修改指南，非常方便二次修改。

## 主要特色

1.包含86版五笔、全拼。

2.按照大陆简体习惯进行了修改和优化，不需任何处理，直接输入简体中文；词频文件换成了大陆语料库提取后的词频，更适合大陆用户习惯。

3.从原始安装包提取的便携版本，采用主流压缩程序制作成自解压程序包，双击即可部署。安装部署参照原版，大幅度优化后，写成纯文本格式的批处理脚本代码，简单操作，确保明码无坑，安全放心，也便于修改。默认安装路径改到D:盘，用户数据安全，不易丢失。

4.五笔状态下支持临时拼音，按z后再输入拼音即可；全拼状态下按“`”引导键，再直接输入单词即可输出英文单词的中文释义。

5.附带详尽的说明文档。除了网上现有的文档外，专门按照修改目的任务为向导整理出来的修改指南，只需搜索要修改的关键词即可快速找到修改方法，DIY定制无比方便。

6.提供方便、灵活、强悍的符号五笔输入方法，一种符号可多种方式输入；

7.可快捷输入当前日期、时间、星期、农历，自动进行数字大小写转换；

10.支持快捷调用外部程序，如命令行、Office软件、截图工具（已经整合）、手写工具、打开特定网页等。

## 安装

使用WinRAR生成的自解压压缩包，双击后会自动解压并启动部署程序。如果已经解压，可双击“!绿化l!.cmd”启动部署程序。之后请在选项中选择所需的方案即可。

注意：多数劣质杀软病毒会把自解压格式的压缩包和脚本代码一律当成病毒。相信杀软的请切勿使用。

## 输入法调用热键

Ctrl+Shift+1。

热键修改方法：控制面板→时钟和区域→区域→格式→语言首选项→拼写、键入和盘符设置→高级键盘设置→输入语言热键，选中要设置的输入法，点击更改按键顺序，然后指定自己需要的快捷键。或者在cmd命令行修改注册表如下：

rem 设置小狼毫调用热键为 Ctrl+Shift+1

reg add "HKLM\SOFTWARE\Microsoft\CTF\TIP\{A3F4CDED-B1E9-41EE-9CA6-7B4D0DE6CB0A}\LanguageProfile\0x00000804\{3D02CAB6-2B8E-4781-BA20-1C9267529467}" /v "Enable" /t REG\_DWORD /d "1" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\TIP\{A3F4CDED-B1E9-41EE-9CA6-7B4D0DE6CB0A}\LanguageProfile\0x00000804\{3D02CAB6-2B8E-4781-BA20-1C9267529467}" /v "Enable" /t REG\_DWORD /d "1" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\DirectSwitchHotkeys\00001001" /v "CLSID" /t REG\_SZ /d "{A3F4CDED-B1E9-41EE-9CA6-7B4D0DE6CB0A}" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\DirectSwitchHotkeys\00001001" /v "LangId" /t REG\_SZ /d "00000804" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\DirectSwitchHotkeys\00001001" /v "Modifiers" /t REG\_SZ /d "0000c006" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\DirectSwitchHotkeys\00001001" /v "PreservedKeyId" /t REG\_SZ /d "{77AFA2B6-402B-11EA-B282-D8CB8A6ECFD3}" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\DirectSwitchHotkeys\00001001" /v "Profile" /t REG\_SZ /d "{3D02CAB6-2B8E-4781-BA20-1C9267529467}" /f >nul 2>nul

reg add "HKCU\Software\Microsoft\CTF\DirectSwitchHotkeys\00001001" /v "VirtualKey" /t REG\_SZ /d "00000031" /f >nul 2>nul

临时中英切换：左Shift。修改：记事本打开default.custom.yaml，修改“Shift\_L: commit\_code”为自己需要的键。

输入法码表选择菜单：Ctrl+` (即~)。修改方法：用记事本打开default.custom.yaml，修改“hotkeys:”下面的热键。

五笔输入法的临时拼音：按z，然后紧跟后面输入拼音即可。五笔状态下也可直接输入拼音，但不支持整句。依赖于luna\_pinyin.schema.yaml，如果此文件缺失或五笔方案文件wubi86.schema.yaml中“reverse\_lookup:”对它的引用不正确，则无效；如果不需要拼音干扰正常输入，可对五笔方案文件wubi86.schema.yaml中“reverse\_lookup:”下的拼音字典引用文件名进行修改，随意改成不存在的文件的名称即可。

## 深山红叶五笔特殊符号输入方法

编码原则：取小优先，能借则借，末Z识别。

取小优先：符号信息量少，因此按照最小化外观部件拆分原则拆分。比如：÷，拆为：点横点，此时只有三码，按照“末Z识别”原则，末尾重复最末编码（点），或者加z，即：点横点点，或：点横点z。

能借则借：即借用现成的相同或近似的字母。如ā，拆为：横a，不足四码，末尾重复末编码或加z进行识别，即：横aa，或：横az。其他如?、‰等，小圈就借用字母O为编码。

末Z识别：前面已经谈及它的含义，它包含两个含义：末，即重复末编码；Z，即不足四码打Z识别码。

考虑到符号没有明显笔顺的特殊性，词库中对编码采取了容错处理，比如?，可以是：OCC或OCZ，即先要圈再打里面内容；也可以倒过来，COO或COZ。

有些不方便编码的符号集，可使用它的俗称名称的五笔编码，比如文档中要用的编号，就按照“编号”输入，可打出“①②③④⑤⑥⑦⑧⑨⑩??????????”等符号集。

拼音用户也可参照此种编码模式，按照最小优先的原则，以拼音的点-d、横-h等方式制作拼音的符号编码，以方便自己的输入。

例：↑ 折竖z ＜ 撇捺z ♀ O横竖z ↙ 撇折z

## 当前日期、时间输入方法

可输出多种格式的日期，按需选用。

所有输入方案状态下均同时支持五笔、拼音的全拼和简拼、英文方式输入。

五笔：以五笔编码输入：当前日期、日期、年月日均可；

拼音：以日期、当前日期、年月日、星期、农历的简拼 rq dqrq nyq xq、nl等输入均可；

英文：以 date week now输入均可。

## 符号通用快捷输入

五笔状态下可以按照前述方法直接打出特殊符号。

另外，五笔码表对特殊符号使用常用名称进行了五笔编码，为避免与某些常用词发生重码，当通过名称编码打不出来时，可添加“符号”二字在名称后面进行编码，比如“办公符号”等等。需要本套五笔词库的支持，换其他词库则不保证实现。

所有输入法状态下，输入半角 / 号作为引导符，后面跟上符号类型的拼音缩写，即可输入。比如：/sb 可输出上标符号、/lm 可输出罗马数字，等等。

/fh 符号 /dn 电脑 /xq 象棋 /mj 麻将 /sz 色子 /pk 扑克 /bq 表情

/tq 天气 /yy 音乐 /lx 两性 /bg 八卦 /tt 天体 /xz 星座 /xh 星号

/fk 方块 /jh 几何 /sx 数字 /szq 数字圈 /zmq 字母圈 /zmh 字母弧 /0等 各种数字

/szm 苏州码 /lm 罗马数字 /lmd 大写罗马 /sb 上标 /xb 下标 /xl 希腊字母 /xld 大写希腊

/ey 俄语 /eyd 大写俄语 /yf 月份 /rq 日期 /yr 曜日 /sj 时间 /tg 天干

/dz 地支 /jq 节气 /dw 单位 /hb 货币 /jg 结构 /pp 偏旁 /kx 康熙部首

/bh 笔划 /bd 标点 /py 拼音 /hzq 汉字圈 /hzh 汉字弧 /sd 声调 /jm 假名

/hw 韩文

## Z键扩展简码

考虑到Z键没有充分利用，因此将比较常用但不易组词的四码高频字“被”、“使”、“都”设定在Z键，即“被 z”、“使 zz”、“都 zzz”

你们 我们 他们 她们分配到z，比如 你们，只取你的第一码，再加z。

## 容错码

记住几个特殊容错，对极常用的高频词进行特殊编码，可极大提升效率。

容错码与正常编码同时存在，即使你不使用容错方式输入，对于正常使用没有任何影响。

1.“狐狸 猩猩”等带有反犬旁词组的严重重码词，我们不对左侧反犬进行二字根拆分，而是视为一个字根，再取右侧。

2.“不要 还要”等常用词重码，我们把“还”反拆，即“之”“不”这样拆，就可将这组最常用的重码分开。

3.“问题”极常用，但重码多，带有这二字的四字词也极多重码。因此我们可把“问”里面的“口”象形地改为字母“O”，即可消除重码。

4.“通”组成的词，以“之”充当首码，这样,“交通 不通 通用 不通 通过 通行 通常 通告 通气”等极多的词组也不重码了。

4.常用连词的容错编码。由于连词在句子中没有具体含义，但交通却起到承上启下、连接词组构成句子的作用，因此它们很常见、也很中性和通用。我们把“因为 所以 不但 而且”等最常用的连词，每字取各取1码，再加z。熟练之后，非常流畅！

## 编辑词库

使用记事本等文本编辑器，打开输入法目录下的 data\wubi86.dict.yaml等带有 dict 字样的文件，参照原有内容的格式修改或添加。

## 编辑扩展词库

使用记事本等文本编辑器，打开输入法目录下的 wubi86.extended.dict.yaml等带有 extended 字样的文件，参照原有内容的格式修改或添加。

词语不需人工编码。注意如果词组中含有一级简码的字，输入时要是一级简码编码，而不能再拆分。比如含有“以”时，编码应当为c，而不是n。

注意：短语中如果存在标点符号则可能无效！

此功能非常适合导入单位的人员名单等，只需把名单一个一行粘贴进去，不需人工编码，然后就能够自动生成编码打字了！单位办公不要太方便！

也可用于维护专业特殊词汇。

## 添加用户自定义短语

使用记事本等文本编辑器，打开输入法目录下的 data\custom\_phrase.txt，参照示例文本进行添加。编码自己随意定义；短语支持标点符号。

## 小狼毫常用参数修改指南

本修改指南的每个修改条目尽量以多个关键词作为条目标题，你可以使用简短的关键词进行文本搜索，比如“候选”、“F4”、“顶屏”、“快捷键”等，即可快速定位到要修改的具体方法内容。可多次搜索以找到尽可能全面的内容。

修改参数注意事项

小狼毫的方案配置分三个级别：一是程序目录下DATA目录中default.yaml、key\_bindings.yaml等几个不包含具体输入法方案名称的文件，这些是针对整个输入法的配置；二是XXX.schema.yaml，其中XXX代表具体的输入方案名称，如wubi86.custom.yaml、luna\_pinyin.schema.yaml等，这些配置专门针对具体的某个输入法的设置；三是中间带有 custom 字样的文件，一般要放到用户数据目录中，作用是对前面两类文件的内容的补充和修正，通常以patch:段来定义参数设置，它的存在可确保程序升级时不会因为程序原有的配置文件的更新而丢失用户修改好的配置，同时便于用户对这些带有 custom 字样的文件进行参数设置备份。

方案配置文件中，程序目录的原始配置文件一般都会全局生效，同时升级后这些配置文件可能会使新版本覆盖。因此，除非你愿意备份，否则建议修改用户数据目录中的配置文件。用户目录中的配置文件通常中间带有 custom 字样。

以下指南中的位置和对应的要修改的目标文件，只是示例性质，具体要根据需求灵活把握，为叙述方便，具体的输入法方案文件以程序目录中的XXX.schema.yaml为参照，此时你也可以修改用户数据目录中对应的XXX.custom.yaml来实现。

对于具体的输入法方案文件，比如XXX.schema.yaml等，可在用户目录创建wubi86.custom.yaml进行修改，但如果你采用的是你自己编写的输入方案，程序升级时不会覆盖它，也可直接对你的方案文件进行修改。以下涉及XXX.schema.yaml的修改方法均可如此，不一一说明。

如果全局配置文件与用户目录中带有custom字样的配置文件内容冲突时，用户目录中带有custom字样的配置文件内容优先。

修改配置后不能立即生效。要让修改结果使应用，请在系统托盘的小狼毫输入法图标上右击鼠标，在弹出的菜单中点击“重新部署”。个别情况下，重新部署后也可能不生效，此时可备份用户数据，在系统托盘图标的快捷菜单中结束小狼毫算法服务、再清除数据目录，之后再重新启动算法服务、重新部署。

如果修改后不能生效，除了重新部署外，则请检查：1.用户目录中的含有custom字样的配置与程序目录中的全局配置是否冲突？2.修改的段的位置是否正确？一定要确保修改的参数处于正确的段名下面。3.修改的参数的空格是否规范？yaml格式的文件对空格非常敏感，规则在此不多说，请参照既有文件内容的空格，让你修改的参数的空格与它们一样对齐即可；特别要注意冒号是半角的，且冒号后面要有一个空格！4.是否在自定义的配置文件中存在多个patch:段？如果有，多余的一定要删除。5.patch:要顶格，其他的段名要用空格缩进。

所有带有custom字样的配置文件中，要以补丁方式即以patch:段的方式添加自定义内容。注意一个配置文件中只能包含一个patch:段。

建议在修改或删除原有内容时，不要直接删除，而是将不需要的行前面添加#号注释掉，让原有的这行不生效，以方便日后修改和查对参考。

用户数据目录的默认位置在“c:\Users\你当前的系统登录账户名\AppData\Roaming\Rime”具体根据实际部署的路径而定。下同。

设置候选词个数、菜单项数目、选词数、候选条目数、候选项数

位置：用户目录，文件default.custom.yaml（如果没有则新建一个UTF-8格式文本文件）。

设置内容：

patch:

"menu/page\_size": 8

然后进行重新部署。

设置配色方案、是否显示托盘图标、候选栏横排还是竖排显示、显示字体、是否显示系统托盘的输入法图标

位置：用户目录，weasel.custom.yaml。

patch:

"style/color\_scheme": google\_plus #配色方案

"style/display\_tray\_icon": false #是否显示系统托盘的[中]或[A]图标

"style/font\_face": "Microsoft YaHei" #显示字体

"style/font\_point": 13 #显示字体大小

"style/horizontal": true #候选栏横排或竖排

"style/display\_tray\_icon": false #是否显示系统托盘的输入法图标

输入默认英文

位置：程序目录，与输入方案对应的.schema.yaml文件。

switches:

- name: ascii\_mode

reset: 1 # 1为默认英文，0为默认中文

针对特定程序自动进入英文模式

位置：用户目录，weasel.custom.yaml

patch:

app\_options:

gvim.exe:

ascii\_mode: true

cmd.exe:

ascii\_mode: true

注意文件里只能有一个patch, 如果存在就在其下追加即可。

设置为默认简体、繁简转换、繁体简体、中英文标点转换、中文标点、英文标点

位置：程序目录。方案文件 XXX.schema.yaml。

switches:

- name: ascii\_mode

reset: 0

states: ["中文", "西文"]

- name: full\_shape

states: ["半角", "全角"]

- name: simplification

reset: 1 #1为简体。reset可不写，此时切换窗口时不会重置到默认状态

states: ["漢字", "汉字"]

- name: ascii\_punct #中英文标点转换开关，0为中文句读，1为英文标点。

states: ["。，", "．，"]

注意，添加或修改 - name: simplification 下面内容为 reset: 1

修改输入法切换快捷键、修改默认的F4快捷键、修改热键、修改组合键、修改功能键

位置：程序目录。修改weasel.yaml。默认为F4，极易与其他用途冲突，可修改。比如改为：

"switcher/hotkeys":

- "Control+grave"

grave即 ~ 号对应的键位。

支持设置多个快捷键以适合不同习惯。

中英文切换快捷键、修改SHIFT键、英文模式

位置：用户数据目录 default.custom.yaml；或程序目录default.yaml。

设为 noop, 屏蔽该切换键

如果要把Caps Lock 设为只改变字母的大小写而不做中西文切换，可将 Caps\_Lock 对应的切换方式设为 noop

如果要以Caps Lock 切换到西文模式，默认输出小写字母，请置 ascii\_composer/good\_old\_caps\_lock: false

如果要以Caps Lock 切换到西文模式，默认输出大写字母，请置 ascii\_composer/good\_old\_caps\_lock: true

ascii\_composer/good\_old\_caps\_lock: false

ascii\_composer/switch\_key:

Caps\_Lock: noop

Shift\_L: commit\_code

Shift\_R: noop

Control\_L: noop

Control\_R: noop

自动清除空码

自动清除空码可在编码输入错误、并且达到预定的编码最大长度时，自动清除空码，以方便重新输入，而不必删除后再重输入。通常五笔方案中比较实用。

位置：程序目录的输入法方案文件 XXX.schema.yaml。

key\_binder:

bindings:

在此段下面添加以下两行，并配合本方案配置文件中的的enable\_sentence: false、auto\_clear: max\_length（如果没有则添加到custom\_phrase: 段中），实现自动清除空码。

- {when: has\_menu, accept: space, send: space}

- {when: composing, accept: space, send: Escape}

自动上屏、设置最大码长、自动清除空码、无重码自动上屏

位置：程序目录的输入法方案文件 XXX.schema.yaml。

speller:

delimiter: " ;'"

auto\_select: true #自动上屏

max\_code\_length: 4 #最大码长

auto\_clear: max\_length #自动清除空码

auto\_select\_unique\_candidate: true #无重码自动上屏

设置扩展词库、用户词库、词库格式、编码逐渐提示（中间显示后续编码）、字符集、整句连打（整句输入、句子输入模式）、自动造词和造词长度、四码自动上屏、用户词典、字频与词频、三码以下不自动调频（解决单字固顶与造词的逻辑冲突）

位置：程序目录的输入法方案文件 XXX.schema.yaml。

translator:

dictionary: wubi86.extended #字典文件

user\_dict: wb86 #在本地生成的用户词典名称

db\_class: tabledb #用户词典格式，tabledb（文本） 或 userdb （二进制）

initial\_quality: 0.5 #该翻译器出字优先级别

enable\_completion: true #编码逐渐提示

enable\_charset\_filter: true #字符集过滤，低重形码用不着

enable\_sentence: false #是否整句连打，是否自动造句，否则为四码自动上屏

enable\_encoder: true #是否自动造词

encode\_commit\_history: true #对已上屏的内容整合成词条，看需求

max\_phrase\_length: 10 #自动造词的最长字数

enable\_user\_dict: true #是否开启用户词典，以记录动态字频和用户词词频

disable\_user\_dict\_for\_patterns:

# - "^z.\*$" #这是原始默认设置

- "^[a-y]{1,3}$" #三码及以下不使用自动调频、不自动造词

comment\_format:

- xform/^~//

- xform/^([a-zA-Z]{4,})/✽/

自动造词

位置：程序目录的输入法方案文件 XXX.schema.yaml。

translator:

# 开启自动造词相关设置，将下面四项都改为 true

enable\_sentence: false # 句子输入模式

enable\_user\_dict: true # 是否开启用户词典（用户词典记录动态字词频，用户词）

enable\_encoder: false # 自动造词

encode\_commit\_history: false # 是否对已上屏的词自动造词

max\_phrase\_length: 10 # 自动生成词的最大长度

user\_dict: user # 用户词典名

db\_class: tabledb # 用户词典类型 userdb - 二进制 / tabledb - 人类语言

disable\_user\_dict\_for\_patterns: # 不需要录入用户词典的编码

- "^z.\*$"

speller:

# 如果想要开启自动造词功能，把下面三项都注释掉，在前面添加 # 这个符号即可

max\_code\_length: 4 # 四码上屏

auto\_select: true # 自动上屏

auto\_select\_unique\_candidate: true # 无重码自动上屏

# alphabet: zyxwvutsrqponmlkjihgfedcba/ # 定义可参与编码的字母表

自定义短语、魔法字符串

与词库和扩展词库不同，自定义短语支持较长的句子，支持句子包含标点符号，通常用于快速定义单位信息、电子邮箱地址以及常用的较长句子等。

位置：1.用户目录，创建UTF-8格式的custom\_phrase.txt短语文件，格式与主词库相同，即“文字、编码、权重（决定重码的次序、可选）”，码表各字段以制表符（Tab）分隔。

2.程序目录的输入法方案文件 XXX.schema.yaml。进行如下修改：

engine:

translators:

- table\_translator@custom\_phrase #使用自定义用户短语，与下面custom\_phrase: 配合使用

以下顶格，不要放在其他段的下面：

custom\_phrase: #使用自定义用户短语，与translators/- table\_translator@custom\_phrase配合；

dictionary: ""

user\_dict: custom\_phrase

db\_class: stabledb

enable\_completion: false

enable\_sentence: false

initial\_quality: 1

设置标点符号由符号文件symbols.yaml统一调用

位置：程序目录的输入法方案文件 XXX.schema.yaml。

punctuator:

import\_preset: symbols #由外部统一文件导入

定义码元集（即允许参与编码的字母）

位置：程序目录的输入法方案文件 XXX.schema.yaml。

speller:

alphabet: zyxwvutsrqponmlkjihgfedcba

设置拼音模糊音定义

位置：程序目录的输入法方案文件 XXX.schema.yaml。

speller:

# 模糊音定义

- derive/^([zcs])h/$1/ # zh, ch, sh => z, c, s

- derive/^([zcs])([^h])/$1h$2/ # z, c, s => zh, ch, sh

# 韵母部份

- derive/([ei])n$/$1ng/ # en => eng, in => ing

- derive/([ei])ng$/$1n/ # eng => en, ing => in

- derive/([a])n$/$1ng/ # an => ang

- derive/([a])ng$/$1n/ # ang => an

- derive/([iu])an$/$1ang/ # ian => iang, uan=uang

- derive/([iu])ang$/$1an/ #iang =>ian, uang=>uan

# 模糊音定义先于简拼定义，方可令简拼支持以上模糊音

- abbrev/^([a-z]).+$/$1/ # 简拼（首字母）

- abbrev/^([zcs]h).+$/$1/ # 简拼（zh, ch, sh）

- derive/^([nl])ve$/$1ue/

- derive/^([jqxy])u/$1v/

- derive/un$/uen/

- derive/ui$/uei/

- derive/iu$/iou/

- derive/([aeiou])ng$/$1gn/

- derive/([dtngkhrzcs])o(u|ng)$/$1o/

- derive/ong$/on/

- derive/ao$/oa/

- derive/([iu])a(o|ng?)$/a$1$2/

双音键盘映射、键位定义、键位设置

位置：程序目录的输入法方案文件 XXX.schema.yaml。

以下是微软拼音的键位映射完整内容，可参考后改为自己的键位映射方案。

speller:

alphabet: zyxwvutsrqponmlkjihgfedcba;

initials: zyxwvutsrqponmlkjihgfedcba

delimiter: " '"

algebra:

- erase/^xx$/

- derive/^([jqxy])u$/$1v/

- derive/^([aoe].\*)$/o$1/

- xform/^([ae])(.\*)$/$1$1$2/

- xform/iu$/Q/

- xform/[iu]a$/W/

- xform/er$|[uv]an$/R/

- xform/[uv]e$/T/

- xform/v$|uai$/Y/

- xform/^sh/U/

- xform/^ch/I/

- xform/^zh/V/

- xform/uo$/O/

- xform/[uv]n$/P/

- xform/i?ong$/S/

- xform/[iu]ang$/D/

- xform/(.)en$/$1F/

- xform/(.)eng$/$1G/

- xform/(.)ang$/$1H/

- xform/ian$/M/

- xform/(.)an$/$1J/

- xform/iao$/C/

- xform/(.)ao$/$1K/

- xform/(.)ai$/$1L/

- xform/(.)ei$/$1Z/

- xform/ie$/X/

- xform/ui$/V/

- derive/T$/V/

- xform/(.)ou$/$1B/

- xform/in$/N/

- xform/ing$/;/

- xlit/QWRTYUIOPSDFGHMJCKLZXVBN/qwrtyuiopsdfghmjcklzxvbn/

#- abbrev/^(.).+$/$1/

translator:

dictionary: luna\_pinyin

prism: double\_pinyin\_mspy

preedit\_format:

- xform/([aoe])(\w)/0$2/

- xform/([bpmnljqxy])n/$1in/

- xform/(\w)g/$1eng/

- xform/(\w)q/$1iu/

- xform/([gkhvuirzcs])w/$1ua/

- xform/(\w)w/$1ia/

- xform/([dtnlgkhjqxyvuirzcs])r/$1uan/

- xform/0r/er/

- xform/([dtgkhvuirzcs])v/$1ui/

- xform/(\w)v/$1ve/

- xform/(\w)t/$1ve/

- xform/([gkhvuirzcs])y/$1uai/

- xform/(\w)y/$1v/

- xform/([dtnlgkhvuirzcs])o/$1uo/

- xform/(\w)p/$1un/

- xform/([jqx])s/$1iong/

- xform/(\w)s/$1ong/

- xform/([jqxnl])d/$1iang/

- xform/(\w)d/$1uang/

- xform/(\w)f/$1en/

- xform/(\w)h/$1ang/

- xform/(\w)j/$1an/

- xform/(\w)k/$1ao/

- xform/(\w)l/$1ai/

- xform/(\w)z/$1ei/

- xform/(\w)x/$1ie/

- xform/(\w)c/$1iao/

- xform/(\w)b/$1ou/

- xform/(\w)m/$1ian/

- xform/(\w);/$1ing/

- xform/0(\w)/$1/

- "xform/(^|[ '])v/$1zh/"

- "xform/(^|[ '])i/$1ch/"

- "xform/(^|[ '])u/$1sh/"

- xform/([jqxy])v/$1u/

- xform/([nl])v/$1ü/

标点符号键位设置、标点符号修改、顿号、/、符号上屏

位置：1.程序目录symbols.yaml。需要在程序目录的输入法方案文件 XXX.schema.yaml设置punctuator/import\_preset: symbols，即标点符号由符号文件symbols.yaml统一调用。

2.程序目录punctuation.yaml。

其中，full\_shape为全角状态定义，half\_shape为状态定义。

比如， 在想让%键能够直接输出%号，而不是默认的多个符号再选择，可修改：

'%' : [ '%', ％, ‰, '°', '℃' ]

改为：

'%' : '%'

其他比如可以让引号键一次输出一对引号，则改为：

'"' : '“”'

修改特殊符号的输入习惯

位置：程序目录的symbols.yaml。

原文件内容是以/打头，再输入符号类型名称的简拼字母，来实现一类符号的批量输出。比如：

#表情

'/bq': [ ☻, ☺, ☹ ]

如果你不习惯或不喜欢用“表情”来表示，而想用“笑脸”来表示，可复制 '/bq': [ ☻, ☺, ☹ ]，然后改为 '/xlq': [ ☻, ☺, ☹ ]。等等。

修改输入法栏配色方案、修改主题、修改皮肤、Sheme

位置：用户目录，weasel.custom.yaml。如果修改程序目录的weasel.yaml则对所有输入法方案均有效，但注意升级后会被覆盖。

建议可以借助Rime See Me或RimeCtrol之类工具直观地修改。

示例：

patch:

preset\_color\_schemes:

default: #皮肤的ID名

name: "安宁／ Default" #皮肤在设置界面的显示名称

author: "深山红叶 <26297628@qq.com>" #皮肤的作者信息

label\_color: 0x808080 #标签颜色

back\_color: 0xEFFFF8 #候选框背景颜色

text\_color: 0x666666 #选择的文本颜色

candidate\_text\_color: 0x000000 #候选文本颜色

border\_color: 0xC9C9FF #候选框边框颜色

comment\_text\_color: 0x4141A3 #附加注释提示文字颜色

hilited\_text\_color: 0xFF0008 #已选择字右侧拼音 文字颜色

hilited\_back\_color: 0xEDEDED #已选择字右侧拼音 背景色

hilited\_candidate\_text\_color: 0x000000 #候选字颜色

hilited\_candidate\_back\_color: 0xFDD5B4 #候选字背景色

candidate\_text\_color: 0x545454 #未候选字颜色

style:

"display\_tray\_icon": false #是否显示系统托盘[中]或[A]图标

"horizontal": true #横排显示

"font\_face": "黑体" #字体

"font\_point": 13 #字体大小

"inline\_preedit": true # 嵌入式候选窗单行显示

layout:

"border\_width": 0 #边框宽度

"border": 0 #边框

"margin\_x": 8 #候选字左右边距

"margin\_y": 8 #候选字上下边距

"hilite\_padding": 8 #候选字背景色色块高度 若想候选字背景色块无边界填充候选框，仅需其高度和候选字上下边距一致即可

"hilite\_spacing": 3 # 序号和候选字之间的间隔

"spacing": 10 #间隔

"candidate\_spacing": 12 # 候选字间隔

"round\_corner": 0 #候选字背景色块圆角幅度（MAC系统有效，Windows系统暂不支持）

中文状态下, shift键字母直接上屏、修改SHIFT切换键

位置：用户目录，default.custom.yaml

patch:

ascii\_composer/switch\_key:

Shift\_L: commit\_code #左Shift直接大写字母上屏。默认是clear。

Shift\_R: commit\_code #右Shift直接大写字母上屏。默认是commit\_text。

始终输出半角标点符号

在已进入小狼毫输入法的情况下，按Ctrl+grave（Tab上面那个键。具体以实际设置的快捷键为准，请参照前文的快捷键设置）

选择，。->，．即可。

设置用户数据目录的路径、用户目录位置

位置：程序目录。运行WeaselSetup.exe，然后按对话框中指示完成路径设置。

也可通过手工修改注册表来设定用户目录路径：

Reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\WOW6432Node\Rime\Weasel" /f /v "WeaselRoot" /t REG\_SZ /d "你要指定的用户数据目录路径"

建议将用户数据目录路径指向非C:盘，以便重装系统时不会丢失用户数据。

个人词库配置同步、网盘同步、在线同步词库、词库自动备份

位置：用户数据目录，修改installation.yaml:

installation\_id: "XXXX" #自定义一个名字, 方便查看

sync\_dir: 'C:\Users\[username]\AppData\Roaming\Rime\sync' #需要同步的用户数据目录。这里是默认的目录位置示例，具体要以你实际设定的用户目录路径为准。注意这里要用单引号！

重新部署；再点击“用户资料同步”；之后再找个同步的工具，将同步目录与上述了的数据目录对应设置好，即可同步。

五笔输入法配置简入繁出

1. 打开Rime程序目录

2. 复制一份 [wubi86.schema.yaml]文件, 重命名为: [wubi86\_trad.schema.yaml]

并修改此文件中的:

schema\_id: wubi86 ==> schema\_id: wubi86\_trad

name: "五笔86" ==> name: "五笔86·简入繁出"

3. https://bintray.com/byvoid/opencc/OpenCC#files

下载OpenCC, 解压后, 将[wubi86.dict.yaml]复制到 OpenCC解压后的文件夹中,

打开CMD(窗口键+R, 输入cmd回车), 转到当前目录(cd 目录地址 回车, 非当前盘符, 需要先转到当前盘符, 如在D盘, 在命令行中先输入d: 再回车, 再输入cd 目录地址 回车)

输入opencc -i wubi86.dict.yaml -o wubi86\_trad.dict.yaml

4. 再将这个文件[wubi86\_trad.dict.yaml]拷贝到Rime程序目录

5. 用文件编辑器打开

修改name: wubi86 ==> name: wubi86\_trad

6. 打开Rime用户目录

Win7 一般会在这个目录下 x:\Users\xxxxx\AppData\Roaming\Rime

在schema\_list下面添加 - {schema: wubi86\_trad}

请注意空格, 保持格式.

7. 最后, 点击【小狼毫】重新部署。 然后, 按Ctrl+grave（Tab上面那个键）调用 "五笔86·简入繁出"。

鼠标跟随设置、光标跟随

位置：用户目录XXX.custom.yaml文件。但实际上似乎无效，一直跟随鼠标光标。

patch:

style/inline\_preedit: false

回车清屏，分号、引号上屏幕，二三候选词、23候选词

位置：1.针对某个输入法的配置：用户数据目录的输入法自定义方案文件 XXX.coustom.yaml，或程序目录的XXX.schema.yaml。

2.全局性配置：程序目录key\_bindings.yaml。

patch: #如果修改程序目录key\_bindings.yaml，则不要patch:

"key\_binder/bindings":

- { when: composing, accept: Return, send: Escape }

- { when: has\_menu, accept: semicolon, send: 2 }

- { when: has\_menu, accept: apostrophe, send: 3 }

其他需要可参照key\_bindings.yaml内容灵活修改。

句号顶屏、句号上屏

配置：程序目录key\_bindings.yaml。

paging\_with\_comma\_period:

\_\_append:

# - { when: has\_menu, accept: period, send: Page\_Down } #本句用#号屏蔽即可。如果启用，则句号将无法顶屏。

小狼毫只能输入英文、只能打英文、不能打汉字、不能出汉字、打不了字、不能打字、无法使用，其他输入法可以正常输入中文

原因是启动项禁用了小狼毫的算法服务（WeaselServer.exe）。

解决多种：

1.可重装小狼毫；

2.或者在命令行窗口使用以下命令：

reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /f /v "WeaselServer" /t REG\_SZ /d "这里换上小狼毫的实际安装路径\WeaselServer.exe"

3.或者运行cmd.exe进入命令行，进入小狼原有安装目录的路径，再输入以下命令：

WeaselSetup.exe /i

WeaselDeployer.exe /install

WeaselServer.exe

输入法反查、五笔查拼音、反查翻译器、用另一种编码方案查码、反查编码、临时拼音、引导符

位置：程序目录XXX.schema.yaml文件。以某五笔的反查设置为例：

engine:

translators:

- punct\_translator

reverse\_lookup:

dictionary: luna\_pinyin #用于反查编码的另一种输入法词库名称

prefix: "z" #反查引导符，即按下这个键后再输入另一种输入法的编码

suffix: "'"

tips: 〔拼音〕

preedit\_format:

- xform/([nl])v/$1ü/

- xform/([nl])ue/$1üe/

- xform/([jqxy])v/$1u/

comment\_format:

- xform/.\*(\b.+$)/\1/

单字优先

位置：程序目录XXX.schema.yaml文件。以某五笔的反查设置为例：

engine:

filters:

- lua\_filter@single\_char\_first\_filter # 单字优先方法