1. [安装及配置 Mac 上的 Rime 输入法——鼠鬚管 (Squirrel)(也适用于小狼豪)](https://blog.csdn.net/hotdust/article/details/52299542) 万次阅读 2016-08-24 13:12:02
2. 可以参考 default.yaml 和相应 输入法 方案的 配置 （比如 double*pinyin*flypy.schema.yaml） 比如在 default.yaml 中看到 1 2 3 punctuator : half\_shape : " /" : [ " 、" ...

http://www.dreamxu.com/install-config-squirrel/

奉送原甫侍讀出守永興 嘉祐五年 歐陽修

酌君以荊州魚枕之蕉 贈君以宣城鼠鬚之管 酒如長虹飲滄海 筆若駿馬馳平坂 受君尚少力方豪 嗟我久衰歡漸鮮 文章驚世知名早 意氣論交相得晩 魚枕蕉 一舉十分當覆盞 鼠鬚管 爲物雖微情不淺 新詩醉墨時一揮 別後寄我無辭遠

这首诗应该就是「鼠须管」的出处吧

奉上一张用「小鹤双拼·语句流」输入的效果，中英混输 + Emoji 表情哦

安装

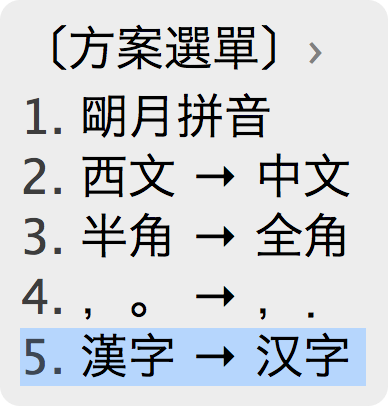
就一条理由就可以使我弃用使用 N 多年的其它输入法 1. https://github.com/lotem/squirrel

Rime 官网：http://rime.im/

可以去官网下载或者使用 Homebrew Cask 来安装（截至到今天版本非最新）

brew cask install squirrel

1. 刚安装好，拼音输入是繁体的，按组合键 Ctrl+` 呼出输入法方案选单（如下），切换为「汉字」就可以输入简体了



## 配置

自行查看《 [Rime 定製指南 | GitHub](https://github.com/rime/home/wiki/CustomizationGuide) 》

以下是我对官网说明书的一些摘录，加上我的一些修改，只限于对拼音输入的理解

如果你用的是小鹤双拼，可以直接看下文《 [详细配置「小鹤双拼·语句流」](http://www.dreamxu.com/install-config-squirrel/#%E8%AF%A6%E7%BB%86%E9%85%8D%E7%BD%AE%E3%80%8C%E5%B0%8F%E9%B9%A4%E5%8F%8C%E6%8B%BC%C2%B7%E8%AF%AD%E5%8F%A5%E6%B5%81%E3%80%8D) 》，介绍的是「小鹤双拼·语句流」的配置。语句流和普通的输入方式在配置上只有一条配置语句的不同，如果不喜欢语句流这种方式，改成普通的也非常简单，下文有介绍。BTW, 现在我自己都不使用「语句流」这种方式了

### 数据文件位置

共享资料夹：”/Library/Input Methods/Squirrel.app/Contents/SharedSupport/”

用户资料夹：”~/Library/Rime/”

### 用户资料夹内的文件说明如下

* 〔全局设定〕 default.yaml
* 〔发行版设定〕 weasel.yaml
* 〔预设输入方案副本〕 <方案标识>.schema.yaml
* **※〔安装信息〕 installation.yaml**
* **※〔用户状态信息〕 user.yaml**

编译输入方案所产生的二进制文件：

* 〔Rime 棱镜〕 <方案标识>.prism.bin
* 〔Rime 固态词典〕 <词典名>.table.bin
* 〔Rime 反查词典〕 <词典名>.reverse.bin

记录用户写作习惯的文件：

* **※〔用户词典〕 <词典名>.userdb.kct**
* **※〔用户词典快照〕 <词典名>.userdb.txt、<词典名>.userdb.kct.snapshot 见于同步文件夾**

以及用户自己设定的：

* **※〔用户对全局设定的定制信息〕 default.custom.yaml**
* **※〔用户对预设输入方案的定制信息〕 <方案标识>.custom.yaml**
* **※〔用户自制输入方案〕及配套的词典源文件**

**注：以上标有「※ 号」和「粗体」的文件，包含用户资料，您在清理文件时要注意备份！**

### 自定义配置

#### 对全局的定制

新建文件 touch ~/Library/Rime/default.custom.yaml, 内容如下

1

2

3

4

5

6

7

8

9

10

# default.custom.yaml, 全局生效

# 添加小鹤双拼到方案选单

patch:

schema\_list:

- schema: double\_pinyin\_flypy # 新增小鹤双拼

- schema: luna\_pinyin

- schema: cangjie5

- schema: luna\_pinyin\_fluency

- schema: luna\_pinyin\_simp

- schema: luna\_pinyin\_tw

#### 对输入方案的定制

比如定制小鹤双拼

新建文件 touch ~/Library/Rime/double\_pinyin\_flypy.custom.yaml

1

2

3

# double\_pinyin\_flypy.custom.yaml, 只对小鹤双拼生效

patch:

translator/preedit\_format: {} # 输入双拼码的时候不转化为全拼码

配置好之后需要「重新部署」：切换到「鼠须管」输入法，点击右上角的的输入法图标，点击「重新部署」

之后就可以用小鹤双拼了

### 外观设定

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

# 适用于【鼠须管】0.9.13+

# 位置：~/Library/Rime/squirrel.custom.yaml

# 用法：想要哪项生效，就删去该行行首的 "#" 字符，但注意保留用于缩进的空格

patch:

# us\_keyboard\_layout: true # 键盘选项：应用美式键盘布局

# 状态通知，适当，也可设为全开（always）全关（never）

# show\_notifications\_when: appropriate

#

# style/color\_scheme: luna # 选择配色方案

# style/horizontal: true # 候选窗横向显示

# style/inline\_preedit: false # 关闭内嵌编码，这样就可以显示首行的拼音

# style/corner\_radius: 10 # 窗口圆角半径

# style/border\_height: 0 # 窗口边界高度，大于圆角半径才有效果

# style/border\_width: 0 # 窗口边界宽度，大于圆角半径才有效果

# style/line\_spacing: 1 # 候选词的行间距

# style/spacing: 5 # 在非内嵌编码模式下，预编辑和候选词之间的间距

# style/font\_face: "Hiragino Sans GB W3" # 字体名称

# style/font\_point: 21 # 字号

# 注：预设的配色方案及代码（指定为 style/color\_scheme ）

# 碧水 - aqua

# 青天 - azure

# 明月 - luna

# 墨池 - ink

# 孤寺 - lost\_temple

# 暗堂 - dark\_temple

# 星际我争霸 - starcraft

# 谷歌 - google

# 晒经石 - solarized\_rock

# 简约白 - clean\_white

### 一些快捷键

#### 输入的时候删除一个音节

⌃ + BackSpace 或 ⇧ + BackSpace

#### 从用户词典中删除误上屏的错词

Shift + Fn + Delete

#### Emacs 风格的编辑键

* ↑： ⌃ + p
* ↓： ⌃ + n
* ←： ⌃ + b
* →： ⌃ + f
* 上页： ⌥ + v
* 下页： ⌃ + v
* 句首： ⌃ + a
* 句末： ⌃ + e
* 回退： ⌃ + h
* 刪除： ⌃ + d
* 清空： ⌃ + g

有时候快捷键可能会用不了，那是因为其它应用首先获取了快捷键，这样输入法就不会影响你对其它软件的控制

### 定制标点符号

可以参考 default.yaml 和相应输入法方案的配置（比如 double*pinyin*flypy.schema.yaml）

比如在 default.yaml 中看到

1

2

3

punctuator:

half\_shape:

"/": ["、", "/", "／", "÷"]

意思是当你输入 “/” 的时候，会出现上面列表中的候选

若你想让 “/键” 直接输出 “、”，可以做如下修改

1

2

3

4

5

6

# double\_pinyin\_flypy.custom.yaml

patch:

punctuator/full\_shape:

"/": "、"

punctuator/half\_shape:

"/": "、"

如果你的用的输入方案导入了 symbols.yaml, 你就可以参考 symbols.yaml 文件来修改了，里面还有各种特殊符号

#### 活用标点创建自定义词组

这还有个用法，你可以将一个键或多个键专门用于输入自定义的词组比如

1

2

3

4

# double\_pinyin\_flypy.custom.yaml

patch:

punctuator/half\_shape:

"+": "+\_+"

当你输入 “+” 的时候就会变成 “+\_+” 了

### 在特定程序里关闭中文输入

在一些程序里很少需要输入中文，于是让鼠须管在这些程序里默认不开启中文输入

要自定义 Mac 应用程序的初始转换状态，首先查看应用的 Info.plist 文件得到该应用的 Bundle Identifier, 通常是形如 com.apple.Xcode 的字符串

例如，要在 Xcode 里面默认关闭中文输入，又要在 Alfred 2 里面恢复开启中文输入，可如此设定：

1

2

3

4

5

# squirrel.custom.yaml

patch:

app\_options/com.apple.Xcode:

ascii\_mode: true

app\_options/com.runningwithcrayons.Alfred-2: {}

修改之前先查看以下 squirrel.yaml 文件，默认设置了一些常用的软件

### 更改切换中西文的快捷键

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

# default.custom.yaml

#

# 可用的按键有 Caps\_Lock, Shift\_L, Shift\_R, Control\_L, control\_R

# Mac 系统上的鼠须管不能区分左、右，因此只有对 Shift\_L, Control\_L 的设定起作用

#

# 已输入编码时按切换键，可以进一步设定输入法中西文切换的形式

# 可选的临时切换策略有三：

# inline\_ascii 在输入法的临时西文编辑区内输入字母、数字、符号、空格等，回车上屏后自动复位到中文

# commit\_text 已输入的候选文字上屏并切换至西文输入模式

# commit\_code 已输入的编码字符上屏并切换至西文输入模式

# 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

patch:

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

ascii\_composer/switch\_key:

Caps\_Lock: noop

Shift\_L: commit\_code

Shift\_R: noop

Control\_L: noop

Control\_R: noop

### 以方括号 “[” 和 “]” 来换页

1

2

3

4

5

6

7

8

9

# default.custom.yaml

patch:

key\_binder/bindings:

- when: paging

accept: bracketleft

send: Page\_Up

- when: has\_menu

accept: bracketright

send: Page\_Down

### 定制每页候选数

1

2

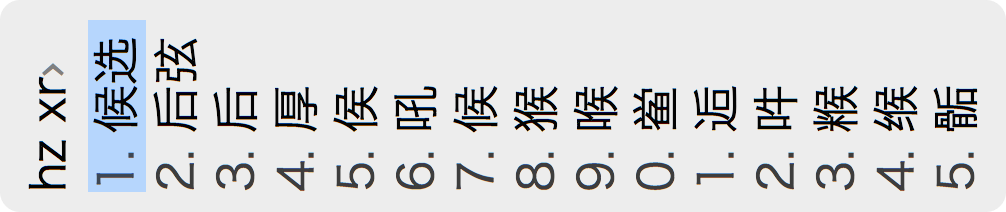
3

# default.custom.yaml

patch:

menu/page\_size: 5

来张 15 个候选的靓照



### 关闭用户词典和字频调整

1

2

3

# double\_pinyin\_flypy.custom.yaml

patch:

translator/enable\_user\_dict: false

### 关于调试

使用过程中难免会出现莫名其妙的问题，这个时候我们可以通过如下方法来查找错误

官方文档关于调试的章节 [https://github.com/rime/home/wiki/RimeWithSchemata#關於調試](https://github.com/rime/home/wiki/RimeWithSchemata#%E9%97%9C%E6%96%BC%E8%AA%BF%E8%A9%A6)

## 词库管理及用户资料的同步

### 用户词库的导入导出

目前鼠须管没有图形工具，使用命令行工具也是一样的

先做一些准备工作

1. 新建文件 touch ~/Library/Rime/rime\_dict\_manager, 输入如下内容

1

2

3

4

5

6

7

8

#!/bin/bash

#

# put this script in ~/Library/Rime, list existing user dictionaries:

# ./rime\_dict\_manager --list

# see other supported options:

# ./rime\_dict\_manager

DYLD\_LIBRARY\_PATH="/Library/Input Methods/Squirrel.app/Contents/Frameworks" "/Library/Input Methods/Squirrel.app/Contents/MacOS/rime\_dict\_manager" $@

1. 然后添加可执行权限

chmod +x ~/Library/Rime/rime\_dict\_manager

1. 进入用户资料夹 (~/Library/Rime/), 执行相关操作

#### 导入导出

1. 查看用户词典列表

./rime\_dict\_manager -l

1. 先关闭输入法，释放以独占方式打开的词典文件

killall Squirrel

1. 将词库导入

./rime\_dict\_manager -i dict\_name import.txt

1. 将词库导出

./rime\_dict\_manager -e dict\_name export.txt

#### 导入扩充词库

应该导入到系统词库，方法：固态词典引用多份码表文件

详细解释请看《 [〔新手推荐敎程〕关于导入词库及「深蓝词库转换」的正确操作方法 | rime 吧](http://tieba.baidu.com/p/2757690418) 》

感谢词库制作者，下载词库点击《 [朙月拼音擴充詞庫 | Rime 翰林院](https://bintray.com/rime-aca/dictionaries/luna_pinyin.dict) 》

导入方法（以「小鹤双拼·语句流」输入方案，extend\_dictionaries20131121.rar 词库文件为例）：

1. 修改定制文件

1

2

3

# double\_pinyin\_flypy\_fluency.custom.yaml

patch:

translator/dictionary: luna\_pinyin.extended

1. 将
2. luna\_pinyin.hanyu.dict.yaml
3. luna\_pinyin.extended.dict.yaml
4. luna\_pinyin.poetry.dict.yaml
5. 这三个词库文件移动到用户资料夹 “~/Library/Rime/”
6. 重新部署
7. **注意** ，在「朙月拼音擴充詞庫」的这个版本中 extend*dictionaries20140909.zip,*他们增加了 *luna*pinyin.cn*en* 这个词库，但是同时他们在 *luna*pinyin.extended.dict.yaml 这个文件写明了双拼不支持 luna*pinyin.cn*en 这个词库，如果你使用的是双拼，记得在 luna*pinyin.extended.dict.yaml* 文件中将 *luna*pinyin.cn\_en 这个去掉
8. 说的更明白一点就是，将 luna\_pinyin.extended.dict.yaml 文件中的：

1

2

3

4

5

import\_tables:

- luna\_pinyin

- luna\_pinyin.hanyu

- luna\_pinyin.poetry

- luna\_pinyin.cn\_en

1. 改写成（去掉 luna*pinyin.cn*en）

1

2

3

4

import\_tables:

- luna\_pinyin

- luna\_pinyin.hanyu

- luna\_pinyin.poetry

### 同步用户资料

#### 使用 Dropbox 同步

编辑 installation.yaml 文件，添加一行：

sync\_dir: '/Users/username/Dropbox/sync/Rime'

#### 在不同电脑间进行同步

同步机制可以理解为这样的：

比如两者的同步文件夹都为 Rime（通过 Dropbox 等同步工具同步），在这之下就会生成两个以 installation\_id 命名的文件夹，分别存放了两者的配置，所以两者之间的配置是不会共享的，自己同步自己的。但是用户词典是相互同步的（显然需要这个功能）

举个同步的例子（我在 Mac 已经进行了相关的配置，也拥有了相当的词库，我又在一台 Windows 上安装了小狼毫输入法，想使用 Mac 下相同的配置和词库）：

1. 在 Windows 上安装小狼毫输入法
2. 使用 Dropbox 等工具将鼠须管的同步文件夹同步到这台 Windows 上（例如为 Y:\Rime）
3. 打开小狼毫的用户文件夹（开始菜单中可直接打开），修改 installation.yaml 文件，添加一行

sync\_dir: Y:\Rime

1. **注意** 不要在路径两边加上双引号，不然无法同步
2. 在开始菜单中点击小狼毫的用户资料同步
3. 将鼠须管的配置文件复制到小狼毫的用户资料文件夹中，例如如下的「小鹤双拼·语句流」的方案，需要复制的配置文件有：
   * double*pinyin*flypy\_fluency.schema.yaml
   * chinese*contain*english.schema.yaml
   * chinese*contain*english.dict.yaml
   * default.custom.yaml
4. 重新部署小狼毫

## 详细配置「小鹤双拼·语句流」

**2014-06-09 更新：**

用了一段时间的语句流，和普通的方式比起来繁琐了一点，但是又没什么明显的优点，至少在 Rime 上是这样子的，我理想中还是觉得语句流要好，但是现在 Rime 还不够强大。

我理想中的语句流输入方式：你输入一系列按键，然后他会自动生成一段话，有时候你会输错，有时候里面夹杂着英文。如果这个输入法非常聪明，他就会正确理解你的意思，就好像你让秘书做事一样，不用告诉他第一步是什么，第二步是什么，他理解你的意思，并且完美地做好。

但是 Rime 做不到秘书这样的水平，你得告诉 Rime 这是中文，这是英文，而且也不能输错。

普通的输入方式：空格上屏，回车输入码直接上屏。相比语句流简单了一点。所以我现在自己也不用语句流了。

虽然当初我写的是语句流的配置，但是没什么关系，他们只有一行配置的不同。详情见 [《创建「小鹤双拼·语句流」输入方案》](http://www.dreamxu.com/install-config-squirrel/#%E5%88%9B%E5%BB%BA%E3%80%8C%E5%B0%8F%E9%B9%A4%E5%8F%8C%E6%8B%BC%C2%B7%E8%AF%AD%E5%8F%A5%E6%B5%81%E3%80%8D%E8%BE%93%E5%85%A5%E6%96%B9%E6%A1%88) 配置文件中的第 40 行。

————————

首先简单介绍一下语句流

玩法是：空格断词，回车和标点上屏

令输入码直接上屏：输入的时候切换为西文编码，然后回车上屏

对用户来说优点很明显，整句输入真的很爽，中英文混输很方便

对输入法来说，它能知道你输入的更多信息，使你输入更爽快

效果图在最上面， [点这里乘电梯](http://www.dreamxu.com/install-config-squirrel/#)

### 创建「小鹤双拼·语句流」输入方案

在「小鹤双拼」的基础上进行修改

cp /Library/Input\ Methods/Squirrel.app/Contents/SharedSupport/double\_pinyin\_flypy.schema.yaml ~/Library/Rime/double\_pinyin\_flypy\_fluency.schema.yaml

不同的地方基本都有说明

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

# double\_pinyin\_flypy\_fluency.schema.yaml

# Rime schema

# encoding: utf-8

schema:

schema\_id: double\_pinyin\_flypy\_fluency # 更改方案标识

name: 小鶴雙拼·語句流

version: "fluency\_0.17"

author:

- double pinyin layout by 鶴

- Rime schema by 佛振 <chen.sst@gmail.com>

description: |

「朙月拼音＋小鶴雙拼」語句流方案。

以空格分詞、標點或回車上屏。

dependencies:

- stroke # 带别名的 translator 所引用的字典部署时不会自动编译

- chinese\_contain\_english # 只有 translator/dictionary: 指定的字典会被编译

- emoji # 因此需要设置输入方案的依赖关系，保证词典可用

switches:

- name: ascii\_mode

reset: 0

states: [ 中文, 西文 ]

- name: full\_shape

states: [ 半角, 全角 ]

- name: ascii\_punct

states: [ ，。, ，． ]

- name: simplification

states: [ 漢字, 汉字 ]

engine:

processors:

- ascii\_composer

- recognizer

- key\_binder

- speller

- punctuator

- selector

- navigator

- fluency\_editor # 语句流必需，编辑器，处理空格、回车上屏、回退键等

# 如果你不喜欢语句流，可将其改为 express\_editor

segmentors:

- ascii\_segmentor

- matcher

- abc\_segmentor

- punct\_segmentor

- fallback\_segmentor

translators:

- punct\_translator

- reverse\_lookup\_translator

- script\_translator

- table\_translator@chinese\_contain\_english # 添加码表式副翻译器

- table\_translator@custom\_phrase # 同上

- table\_translator@emoji # 同上

filters:

- simplifier

- uniquifier

speller:

alphabet: zyxwvutsrqponmlkjihgfedcba

delimiter: " '"

algebra:

- erase/^xx$/

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

- derive/^([aoe])([ioun])$/$1$1$2/

- xform/^([aoe])(ng)?$/$1$1$2/

- xform/iu$/Q/

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

- xform/uan$/R/

- xform/[uv]e$/T/

- xform/un$/Y/

- xform/^sh/U/

- xform/^ch/I/

- xform/^zh/V/

- xform/uo$/O/

- xform/ie$/P/

- xform/i?ong$/S/

- xform/ing$|uai$/K/

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

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

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

- xform/[iu]ang$/L/

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

- xform/ian$/M/

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

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

- xform/[iu]a$/X/

- xform/iao$/N/

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

- xform/ui$/V/

- xform/in$/B/

- xlit/QWRTYUIOPSDFGHJKLZXCVBNM/qwrtyuiopsdfghjklzxcvbnm/

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

translator:

dictionary: luna\_pinyin

prism: double\_pinyin\_flypy\_fluency # 由拼写运算规则产生的拼写映射表的名称

# 去掉了 preedit\_format:, 作用是输入双拼码的时候不转化为全拼码

# 中英文混输的时候，可补全英文，并在英文两边加上半角空格

# 例如：我爱 Rime 输入法

chinese\_contain\_english:

dictionary: chinese\_contain\_english

# 用户自定义词典，可直接编辑文件 "~/Library/Rime/custom\_phrase.txt"

# db\_class: stabledb 指定词典格式为文本码表格式（文字<TAB>编码<TAB>权重（可选））

custom\_phrase:

dictionary: ""

user\_dict: custom\_phrase

db\_class: stabledb

enable\_completion: false

enable\_sentence: false

initial\_quality: 1

# 可直接输入 Emoji 表情编码来输入表情

emoji:

dictionary: emoji

reverse\_lookup:

dictionary: stroke

enable\_completion: true

prefix: "`"

suffix: "'"

tips: 〔筆畫〕

preedit\_format:

- xlit/hspnz/一丨丿丶乙/

comment\_format:

- xform/([nl])v/$1ü/

punctuator:

import\_preset: symbols # 导入 symbols.yaml, 用于输入特殊符号

key\_binder:

import\_preset: default

recognizer:

import\_preset: default

patterns:

number: "^[-+]?[0-9][.:0-9]\*[%]?$"

# 配合特殊符号的输入

punct: "^/[0-9]\*[a-z]\*$"

reverse\_lookup: "`[a-z]\*'?$"

#### 创建 chinese\_contain\_english 输入方案

因「小鹤双拼·语句流」输入方案依赖 “chinese*contain*english” 输入方案，如果不创建的话 chinese*contain*english 词典无法被编译

创建简单版本即可 touch ~/Library/Rime/chinese\_contain\_english.schema.yaml, 内容如下：

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

# chinese\_contain\_english.schema.yaml

# Rime schema

# encoding: utf-8

schema:

schema\_id: chinese\_contain\_english

name: 英文

version: "0.1"

description:

为了配合「小鹤双拼·语句流」输入方案

engine:

processors:

- speller # 把字母追加到编码串

- selector # 选字、换页

- navigator # 移动插入点

- express\_editor # 空格确认当前输入、其他字符直接上屏

segmentors:

- abc\_segmentor # 标记输入码的类型

translators:

- echo\_translator # 没有候选项时，创建一个与编码串一个模样的候选项

- table\_translator # 码表式转换

translator:

dictionary: chinese\_contain\_english # 设定 table\_translator 使用的词典名

#### 创建 chinese\_contain\_english 词典

touch ~/Library/Rime/chinese\_contain\_english.dict.yaml

内容如下：

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

# chinese\_contain\_english.dict.yaml

# Rime dictionary

# encoding: utf-8

#

# 中英文混输的时候有用

---

name: chinese\_contain\_english

version: "0.1"

sort: by\_weight

use\_preset\_vocabulary: false

...

Rime rime

Android android

Apple apple

Mac mac

Windows windows

# ... 自己扩展

需要 **注意** 的是： Rime rime 之间要有 tab，其它的类似，词典格式是如此

完整的文件内容请看： <https://gist.github.com/lifenod/931e306e2ae8bd32b169>

### 添加「小鹤双拼·语句流」输入方案以及对全局的定制

首先创建对全局的用户定制文件

touch ~/Library/Rime/default.custom.yaml

然后编辑该文件

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

# default.custom.yaml

# Rime default settings

# encoding: utf-8

patch:

schema\_list:

- schema: double\_pinyin\_flypy\_fluency # 新增「小鶴雙拼·語句流」

- schema: luna\_pinyin

- schema: cangjie5

- schema: luna\_pinyin\_fluency

- schema: luna\_pinyin\_simp

- schema: luna\_pinyin\_tw

key\_binder/bindings: # 以方括号换页

- when: paging

accept: bracketleft

send: Page\_Up

- when: has\_menu

accept: bracketright

send: Page\_Down

ascii\_composer/good\_old\_caps\_lock: true # 中西文切换快捷键

ascii\_composer/switch\_key:

Caps\_Lock: noop

Shift\_L: inline\_ascii

Shift\_R: noop

Control\_L: noop

Control\_R: noop

### 外观设定

见上文《 [外观设定](http://www.dreamxu.com/install-config-squirrel/#%E5%A4%96%E8%A7%82%E8%AE%BE%E5%AE%9A) 》

### 导入扩充词库

见上文《 [导入扩充词库](http://www.dreamxu.com/install-config-squirrel/#%E5%AF%BC%E5%85%A5%E6%89%A9%E5%85%85%E8%AF%8D%E5%BA%93) 》

### 同步用户资料

见上文《 [同步用户资料](http://www.dreamxu.com/install-config-squirrel/#%E5%90%8C%E6%AD%A5%E7%94%A8%E6%88%B7%E8%B5%84%E6%96%99) 》

## iOS 上的双拼输入法

在 iOS 9 系统下，我使用的是「SPi 双拼输入法 for iPhone」，一款不需要「完全访问」的输入法，界面近乎原生，简单快速，双拼功能也足够使用

官网： <http://spiim.com/>

GitHub 主页： <https://github.com/guoc/spi>, 这款输入法是开源的

[收起]()



[展开全文]()



1. [小狼毫 Rime 输入法任务导向式常用参数修改指南](https://blog.csdn.net/emca/article/details/104827624) 千次阅读 2020-03-12 20:56:34
2. 【深山红叶整理 v20200226】 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... 配置 小狼毫时，请随时注意系统TEMP目录中的警告和错误信息（用记事本打开以 *rime*.weasel 字符串开头的文...

【深山红叶整理 v20200226】

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。

配置小狼毫时，请随时注意系统TEMP目录中的警告和错误信息（用记事本打开以 rime.weasel 字符串开头的文件），对于分析故障极有帮助，很快就能够找到问题的相关线索。

以下内容均从网络收集后，按照任务导向进行了二次整理，以方便查询。

配置好的资源：https://download.csdn.net/download/emca/12244911

修改 Rime 参数注意事项

1.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。

2.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。

3.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。修改时，尽量优先修改用户数据目录中对应的XXX.custom.yaml，其次才是修改XXX.schema.yaml。

4.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。

5.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；没有custom字样的配置文件？你就在用户数据目录新建一个UTF-8格式的文本文件。

6.修改配置后不能立即生效。要让修改结果立即应用，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。

7.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。

8.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。

9.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。

10.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。

11.码表中的格式错误、配置方案中的任何错误，都可能导致部署后无法打字、输入法算法服务自动崩溃等异常。配置文件格式和语法正确与否，可在线检查：http://www.bejson.com/validators/yaml/

12.如果调用的外部程序存在不能保存配置等异常，则可能是权限限制，可将对应的程序属性的兼容性设置为以管理员身份运行。

小狼毫目录结构、文件作用

程序根目录：

WeaselDeployer.exe：方案设置和皮肤配色外观设置

WeaselServer.exe：输入法算法服务

WeaselSetup.exe：安装选项、用户文件夹位置设置

uninstall.exe：卸载程序

data：程序全局数据目录

Data目录：

default.yaml：全局设定

weasel.yaml：发行版设定

essay.txt：字频文件

key\_bindings.yaml：快捷键、按键绑定、键盘功能映射的定义文件

punctuation.yaml：基本标点符号定义文件

symbols.yaml：基本符号和特殊符号定义文件

<方案标识>.schema.yaml：具体的预设输入方案配置，如luna\_pinyin.schema.yaml等。

<方案标识>.dict.yaml：具体的预设输入方案的词库，如luna\_pinyin.dict.yaml等。

<方案标识>.extended.dict.yaml：具体的预设输入方案的扩展、附加词库

用户数据目录：

（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）

installation.yaml：安装信息

custom\_phrase.txt：用户短语词库（需要用户自建）

user.yaml：用户状态信息

default.custom.yaml：全局设定之用户个性化补充配置设定文件

rime.lua：功能滤镜（需要用户另行创建，原版不附带。通常用于自动输出多格式日期、大小写自动转换等等功能的实现）

<方案标识>.custom.yaml：具体的预设输入方案之用户个性化补充配置

用户数据目录\build目录：

1.编译输入方案所产出的二进制文件

<方案标识>.prism.bin：Rime 棱镜

<词典名>.table.bin：固态词典

<词典名>.reverse.bin：反查词典

2.记录用户写作习惯的文件

※ <词典名>.userdb.kct：用户词典

※ <词典名>.userdb.txt、<词典名>.userdb.kct.snapshot：见用户词典快照，于同步文件夹

3.用户修改化设置

※ default.custom.yaml：用户对全局设定的定制信息

※ <方案标识>.custom.yaml：用户对预设输入方案的定制信息

※ 其他如〔用户自制输入方案〕及配套的词典源文件等。

注：以上标有 ※ 号的文件，包含用户资料，您在清理文件时要注意备份！

设置候选词个数、菜单项数目、选词数、候选条目数、候选项数、候选框、候选窗口

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

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、横竖排、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

输入时默认英文模式、默认不输入中文、默认字母

位置：程序目录，与输入方案对应的.schema.yaml文件，或者用户数据目录中对应的XXX.custom.yaml。

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、自动英文、自动切换英文

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、简繁转换、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，或者用户数据目录中对应的XXX.custom.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、~键、`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、SHIFT上屏、直接上屏、SHIFT大写字母

位置：用户数据目录 default.custom.yaml；或程序目录default.yaml，或者用户数据目录中对应的其他输入法XXX.custom.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 #按SHIFT加字母后，字母直接上屏

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，或者用户数据目录中对应的XXX.custom.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，或者用户数据目录中对应的XXX.custom.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，或者用户数据目录中对应的XXX.custom.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 #编码逐渐提示开关；编码提示

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、用户词、自动组词、在线造词

位置：程序目录的输入法方案文件 XXX.schema.yaml，或者用户数据目录中对应的XXX.custom.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、短句子、习惯用语、个性短语、个性符号、符号定义

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，或者用户数据目录中对应的XXX.custom.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，或者用户数据目录中对应的XXX.custom.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）、可用字母

位置：程序目录的输入法方案文件 XXX.schema.yaml，或者用户数据目录中对应的XXX.custom.yaml。

speller:

alphabet: zyxwvutsrqponmlkjihgfedcba

设置拼音模糊音定义、方言定义

位置：程序目录的输入法方案文件 XXX.schema.yaml，或者用户数据目录中对应的XXX.custom.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/

双拼键盘映射、键位定义、键位设置、微软双拼、小鹤双拼、自然码、拼音加加、搜狗拼音、小鹤音形

位置：程序目录的输入法方案文件 XXX.schema.yaml，或者用户数据目录中对应的XXX.custom.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、分号、句号、引号、/、符号上屏、引导符

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，或者用户数据目录中对应的XXX.custom.yaml中的punctuation:段。

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：

'"' : '“”'

注意：如果希望使用/号充当符号的引导符，则不要将/定义为单一的中文符号，否则/充当引导符无效！！

修改特殊符号的输入习惯、颜文字、emoji

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、窗口颜色、状态栏、字体、边框

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键加字母直接上屏、修改SHIFT切换键、大写字母

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、用户数据、数据备份

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、数据备份

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、打简出繁、简打繁、输出繁体

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

style/inline\_preedit": true #使用内嵌编码

回车清屏，分号；、引号‘’上屏幕，二三候选词、23候选词、2、3候选词、候选键、上屏键

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default.yaml、key\_bindings.yaml，或者用户数据目录中对应的XXX.custom.yaml。

patch: #如果修改程序目default.yaml、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、自动上屏

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：用户数据目录中对应输入法的XXX.custom.yaml，或程序目录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、字频、词频

位置：用户数据目录中对应输入法的XXX.custom.yaml，或程序目录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

设置拼音词组可以打首字母以简拼方式输入

speller:

algebra:

- "abbrev/^([a-z]).+$/$1/"

- "abbrev/^([zcs]h).+$/$1/"

alphabet: zyxwvutsrqponmlkjihgfedcba

delimiter: " '"

1. [收起]()



1. [展开全文]()



1. [小狼毫(Rime)输入法设置Shift直接上屏英文字符并切换为英文状态方法](https://blog.csdn.net/sdujava2011/article/details/84098971?utm_term=rime%E8%BE%93%E5%85%A5%E6%B3%95%E9%85%8D%E7%BD%AE%E8%8B%B1%E6%96%87%E7%BF%BB%E8%AF%91&amp;utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~all~sobaiduweb~default-0-84098971&amp;spm=3001.4430)
2. 其中L指Left,左的意思。commit-提交。code-代码。 当我们 输入 一段文字未上屏之前,按此键后字符将被将直接上屏,*RIME* 切换为 英文输入 状态。再次按此键 *RIME* 切换回中文 输入 状态。 Shift*R: inline*ascii代码: ...

其中L指Left,左的意思。commit-提交。code-代码。 当我们输入一段文字未上屏之前,按此键后字符将被将直接上屏,RIME切换为英文输入状态。再次按此键RIME切换回中文输入状态。 Shift\_R: inline\_ascii代码: ...

1. [RIME输入法配置\_Zerokas的博客\_rime输入法配置](https://blog.csdn.net/e15273/article/details/79567626?utm_term=rime%E8%BE%93%E5%85%A5%E6%B3%95%E9%85%8D%E7%BD%AE%E8%8B%B1%E6%96%87%E7%BF%BB%E8%AF%91&amp;utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~all~sobaiduweb~default-1-79567626&amp;spm=3001.4430)
2. *RIME*输入法配置 <http://tieba.baidu.com/p/3773090625> <https://www.jianguoyun.com/p/DZISfzQQv_3jBRj4zgk> 根据第二篇文章的配置部署没有成功,主要原因就是除了patch是顶格写的,其它有自己的缩进的格式,第一层缩进的值需要用引号...

RIME输入法配置 http://tieba.baidu.com/p/3773090625 https://www.jianguoyun.com/p/DZISfzQQv\_3jBRj4zgk 根据第二篇文章的配置部署没有成功,主要原因就是除了patch是顶格写的,其它有自己的缩进的格式,第一层缩进的值需要用引号...

1. [Rime (简体中文)](https://blog.csdn.net/weixin_33924571/article/details/116822963) 2021-05-15 05:29:40
2. 安装 配置 Note: *Rime* 与IBus (简体中文)和Fcitx (简体中文)了解更多有关安装和 配置 的信息。在使用 *Rime* 之前，需要安装输入方案。输入方案可由用户定制。默认输入方案如下：现代标准汉语：朙月拼音地球拼音(带声调)注音...

翻译状态：本文是 Rime 的翻译。上次翻译日期：2020-01-16。如果英文版本有所更改，则您可以帮助同步翻译。

Fcitx和IBus等输入法框架。

安装

配置

Note: Rime与IBus (简体中文)和Fcitx (简体中文)了解更多有关安装和配置的信息。

在使用Rime之前，需要安装输入方案。输入方案可由用户定制。默认输入方案如下：

现代标准汉语：

朙月拼音

地球拼音(带声调)

注音符号

拼音变体：

双拼(自然码、微软拼音)

方言：

粤拼

吴语

字形输入法：

仓颉

五笔

你可以在任何时间通过按下 F4 调出菜单来切换输入法。参见选择输入方案。

要自定义 Rime，首先需要创建 rime 的配置文件夹。

如果你在使用

$ mkdir ~/.config/ibus/rime

$ mkdir ~/.config/fcitx/rime/

$ mkdir ~/.local/share/fcitx5/rime/

在文件夹中创建 default.custom.yaml 文件，以便于指定可选的输入法。例如，如果你想要按照声调来输入拼音，你可以通过添加下列内容来使用地球拼音：

default.custom.yaml

patch:

schema\_list:

- schema: terra\_pinyin

注意：缩进级别是很重要的。这个文件会覆盖默认的配置文件，因此如果只添加了地球拼音，那么将只有这一个选择。

要使得自定义生效，需要进行重新部署。如果你在使用 ibus 或 fcitx 的 图形界面，你可以找到 ⟲ (部署) 按钮。

或者使用以下的命令：$ rm ~/.config/ibus/rime/default.yaml && ibus-daemon -drx

$ rm ~/.config/fcitx/rime/default.yaml && fcitx-remote -r

注意：声调是可选的，你可以使用它来很好地过滤列表。以下是如何使用：

第一声： -

第二声： /

第三声： <

第四声： \

例如，如果想要打出发音为 hǎo 的字，需要输入 hao< ，输入法会自动将其转换为 hǎo 。

默认情况下，Rime 只会列出5个候选项，可以通过修改 "menu/page\_size" 的值来手动更改列出候选项的个数

default.custom.yaml

patch:

"menu/page\_size": 9

使用

Rime 有一些十分优秀的特质，使之能够很好地输入中文字符和标点

选择输入方案

不论何时，在使用 Rime 的时候，你都可以通过 F4 来进行基础设置。显示的设置项如下：

1.　输入法名称

2. 中文　－›　西文

3. 全角　－›　半角

4. 漢字　－›　汉字

...

第一项显示的是输入方案的名称，如果你开启了多个输入方案，那将能够在其中进行切换

第二项选择输入中文或英语

第三项选择输入全角或半角标点

最后一项选择输入简体中文或繁体中文

中文标点

按下列各键输入不同的符号：

[ -> 「　【　〔　［

] ->　」　】　〕　］

{ ->　『　〖　｛

} ->　』　〗　｝

< ->　《　〈　«　‹

> ->　》　〉　»　›

@ ->　＠　@　☯

/ ->　／　/　÷

\* ->　＊　\*　・　×　※

% ->　％　%　°　℃

$ ->　￥　$　€　£　¥

| ->　・　｜　|　§　¦

\_ -> ——

\ ->　、　＼　\

^ ->　……

~ ->　〜　~　～　〰

高级

参见

1. [收起]()



1. [展开全文]()



1. [Mac 平台最好用的 五笔输入法 鼠须管 Rime 安装教程](https://blog.csdn.net/KimBing/article/details/88042101) 千次阅读 2019-02-28 19:14:49
2. Mac 鼠须管 *Rime* 输入法 安装五笔 输入法 教程 相关链接 极点五笔方案(github)： <https://github.com/KyleBing/> *rime* -wubi86-jidan *RIME* 官网： https:// *rime*.im/ *RIME* github 地址： <https://github.com/> *rime* *RIME* ...

前言

Rime 是一款跨平台的优秀输入法的内核。 该输入法在不同平台的名字也不同

Windows - 小狼毫 (weasel)macOS - 鼠须管 (squirrel)Linux - 中州韵 (ibus-rime)

目前本人已知的 Rime 在 windows 和 macOS 平台上的词库配置是相同的，只是配置文件的名字不一样。 Rime 输入法的优势在于它高度的可自定义化，不单单可以定义输入法码表，还可以定义输入法翻译码表的方式，标点对应等等等等。 高度自定义的特性也使得入门的门槛比较高一些。如果想自定义方案，需要有一定的编程基础，至少有一定的程序语言基础。

用极点输入法的原因

用久了五笔的都知道，喜欢五笔的因为是五笔的重码少，如果码表太多重码体验就很差了。 好词库的特点是：减少特殊词的数量，增加通用词的频率。

极点版是重码很少的，打起字来很爽，而且对标点的支持也很好。 之前用的 清歌输入法，但该输入法有个弊端，对于我这种前端工程师来说，会在工作中用到数字左边那个键 ~，而清歌输入法把这个键作为临时拼音输入的入口，用起来就各种麻烦。 现在换成 Rime 简直爽翻了，好久没有这么爽的打过字了。

使用前的一些用户习惯说明

control + 0 或者 shift + control + 0 弹出菜单

如果第一个组合键不弹出菜单，就用第二个组合键，如果还不弹出就换个软件，进入输入状态再试 有时候是目前的软件环境屏蔽了这个组合键（如：MWeb 中 control + 0 这个组合键就冲突），换个软件再按就可以了

弹的菜单中第一位的名字就是当前使用的输入法方案，其后跟着的就是该方法中的输入法菜单，有【半角 - 全角】【简 - 繁】等功能菜单，再后面的就是其它可选的输入法方案，对应 default.custom.yaml 中 schema\_list 字段内容

关于【简入繁出】支持

是以切换输入方案的形式实现的，之前用菜单实现有个弊端：在切换应用后繁体输出的设置并没有保留，也就是说不是全局的，以输入方案的形式就可以实现全局繁体输入。 control + 0 呼出菜单，选择极点五笔繁体，目前在 Mac 上测试正常，不知道 Windows 上如何，如果不能实现简入繁出的效果，可能需要安装 OpenCC[链接地址] 库支持，具体不知道怎么操作，因为我配好 schema 后就可以用了，没有安装 OpenCC

安装 鼠须管(macOS)

去 官网下载 下载后按照步骤安装即可

1. 下载 五笔配置文件

也就是当前库，直接下载即可 https://github.com/KyleBing/rime-wubi86-jidian

其中的文件列表有：

.

├── README.md # 当前说明文档

├── numbers.schema.yaml # 输入方案 - 大写数字

├── rime.lua # 可以输出系统变量的函数

├── default.custom.yaml # 自定义一些输入法的功能：标点，二三候选等

├── pinyin\_simp.dict.yaml # 简体拼音码表 - 五笔中拼音输入需要的

├── pinyin\_simp.schema.yaml # 输入方案 - 简体拼音

├── squirrel.custom.yaml # 鼠须管（for macOS）输入法候选词界面

├── wubi86\_jidian.dict.yaml # 极点 - 五笔码表

├── wubi86\_jidian.schema.yaml # 输入方案 - 极点五笔

├── wubi86\_jidian\_user.dict.yaml # 扩展词库 - 用户个人词库

├── wubi86\_jidian\_extra\_brand.dict.yaml # 扩展词库 - 品牌

├── wubi86\_jidian\_extra\_english.dict.yaml # 扩展词库 - 常用英文

├── wubi86\_jidian\_extra\_location.dict.yaml # 扩展词库 - 地名

├── wubi86\_jidian\_extra\_media.dict.yaml # 扩展词库 - 影视名，音乐名

├── wubi86\_jidian\_extra\_people.dict.yaml # 扩展词库 - 名人

├── wubi86\_jidian\_extra\_pro.dict.yaml # 扩展词库 - 专业名词

├── wubi86\_jidian\_extra\_game.dict.yaml # 扩展词库 - 游戏

├── wubi86\_jidian\_pinyin.schema.yaml # 输入方案 - 五笔拼音混输

└── wubi86\_jidian\_trad.schema.yaml # 输入方案 - 五笔简入繁出

### 2. 设置五笔输入法 macOS 鼠须管

1. macOS 上的 鼠须管 配置文件存放目录是 ~/Library/Rime
2. 把上面下载的文件移到该目录中，点击 部署 即可。

放的时候目录结构是这样的：

~/Library/

└── Rime

├── README.md

├── default.custom.yaml

├── numbers.schema.yaml

├── pinyin\_simp.dict.yaml

├── pinyin\_simp.schema.yaml

├── squirrel.custom.yaml

├── wubi86\_jidian.dict.yaml

├── wubi86\_jidian.schema.yaml

├── wubi86\_jidian\_user.dict.yaml

├── wubi86\_jidian\_extra\_brand.dict.yaml

├── wubi86\_jidian\_extra\_english.dict.yaml

├── wubi86\_jidian\_extra\_location.dict.yaml

├── wubi86\_jidian\_extra\_media.dict.yaml

├── wubi86\_jidian\_extra\_people.dict.yaml

├── wubi86\_jidian\_extra\_pro.dict.yaml

├── wubi86\_jidian\_extra\_game.dict.yaml

├── wubi86\_jidian\_pinyin.schema.yaml

└── wubi86\_jidian\_trad.schema.yaml

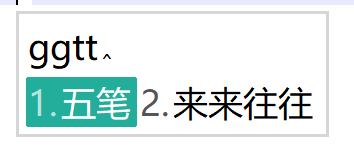
注意：对于不熟悉命令行操作的朋友， ~ 代表的是当前用户的主目录，比如我的用户名是 kyle, ~ 就代表 /Users/kyle/ 这个绝对路径。

需要将你下载的文件放入 /Users/你用户名/Library/Rime 这个目录下，了然否？

### 3. 皮肤



## 配置 小狼毫（Windows）



1. 点击【开始】
2. 打开刚刚安装的小狼毫输入法程序目录，打开【用户文件夹】
3. 把该项目中的文件复制到里面
4. 点击开始菜单中的【部署】即可

## 相关链接

**资源链接**

* 极点五笔方案(github)： <https://github.com/KyleBing/rime-wubi86-jidan>
* Rime github 地址： <https://github.com/rime>
* Rime 输入方案集合： <https://github.com/rime/plum>
* Rime 官方五笔码表： <https://github.com/rime/rime-wubi>
* Rime 简拼输入方案： <https://github.com/rime/rime-pinyin-simp>

**配置教程链接**

* Rime 官网： <https://rime.im/>
* Rime 输入方案参数详解： <https://github.com/LEOYoon-Tsaw/Rime_collections/blob/master/Rime_description.md>
* 中英切换自定义： <https://gist.github.com/lotem/2981316>

## 关于自定义一些功能

所有配置说明都在配置文件中说明了，如果有其它问题可以在 issue 中提出，或者在群里（QQ群：878750538）讨论，有需要就 @青枫

├── numbers.schema.yaml # 输入方案 - 大写数字

├── squirrel.custom.yaml # 鼠须管（for macOS）输入法候选词界面

├── default.custom.yaml # 自定义一些输入法的功能：标点，二三候选等

├── wubi86\_jidian.schema.yaml # 输入方案 - 极点五笔

├── wubi86\_jidian\_user.dict.yaml # 扩展词库 - 用户个人词库

├── wubi86\_jidian\_extra\_brand.dict.yaml # 扩展词库 - 品牌

├── wubi86\_jidian\_extra\_english.dict.yaml # 扩展词库 - 常用英文

├── wubi86\_jidian\_extra\_location.dict.yaml # 扩展词库 - 地名

├── wubi86\_jidian\_extra\_media.dict.yaml # 扩展词库 - 影视名，音乐名

├── wubi86\_jidian\_extra\_people.dict.yaml # 扩展词库 - 名人

├── wubi86\_jidian\_extra\_pro.dict.yaml # 扩展词库 - 专业名词

├── wubi86\_jidian\_extra\_game.dict.yaml # 扩展词库 - 游戏

### 1. 输出系统 时间 和 日期

输入对应词，获取当前日期和时间

* date 输出日期，格式 2019年06月19日 2019-06-19
* time 输出时间，格式 10:00 10:00:00

### 2. 开启五笔模式下的自动造词功能

默认是没有开启的，如果想开启需要手动编辑 wubi86\_jidian.schema.yaml 文件，里面也有相关的说明

除了把文件中自动造词部分都设为 true 之外，还需要把 speller 那段的都注释掉，因为那里都是直接上屏的，直接上屏就无法造词了，所以需要注释掉。

造词功能是这样的，在输入一次之后，输入法会记住这个连词，打的时候后面会有图标指示，下次再输入这个词的时候，就会固定这个词，并在用户词典中新增这个词的词条。

你修改后的配置应该是这样的：

speller:

# 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

translator:

# 开启自动造词相关设置

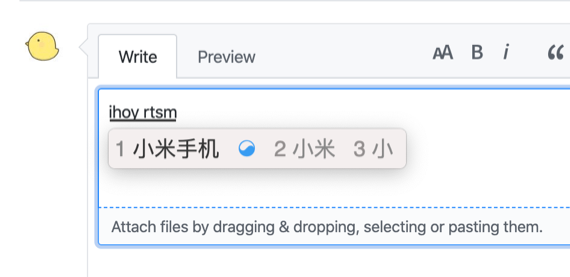
enable\_sentence: ture # 是否开启自动造词

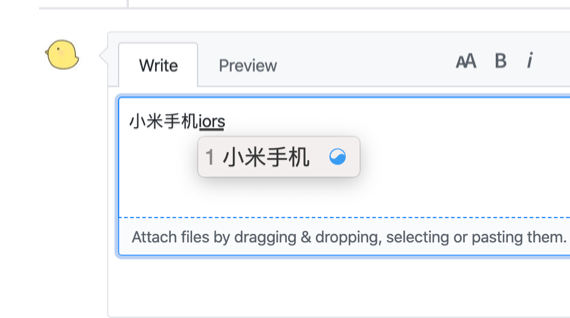
enable\_user\_dict: ture # 是否开启用户词典（用户词典记录动态字词频，用户词）

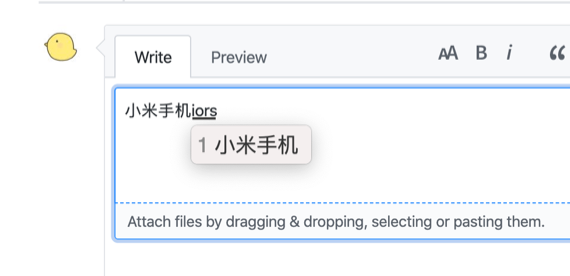
enable\_encoder: ture # 自动造词

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

效果如图：









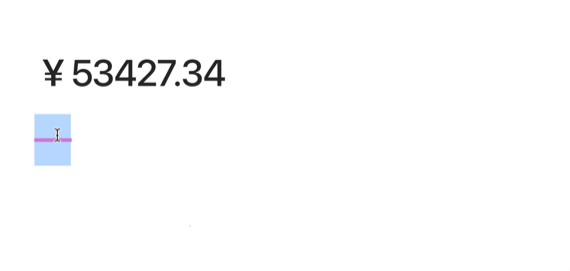
### 3. 快捷输入大写数字：壹贰叁肆伍陆

本库中包含一个可以输入大写数字的方案，名叫 大写数字 ，呼出菜单选择该方案即可。

在这个模式下：具体可以看源文件 [numbers.schema.yaml](https://github.com/KyleBing/rime-wubi86-jidian/blob/master/numbers.schema.yaml)

键对应值1234567890一二三四五六七八九〇wqbsjfd.万千百十角分点z之整y月元亿 键(按住 shift)对应值1234567890壹贰叁肆伍陆柒捌玖零wqbsjfd.万佰仟拾角分第点z之整y月圆亿

输入案例：



### 4. 输出系统变量

自 v0.13 之后可自定义输出系统变量，如日期等

文件 [rime.lua](https://github.com/KyleBing/rime-wubi86-jidian/blob/master/rime.lua) 盛放的是调用的方法，你需要在相应的 XXXX.schema.yaml 文件的 engine / translators 字段添加一些东西，可以参阅本库的 [wubi86\_jidian.schema.yaml](https://github.com/KyleBing/rime-wubi86-jidian/blob/master/wubi86_jidian.schema.yaml) 文件。

具体 rime.lua 文件说明参阅这里： <https://github.com/hchunhui/librime-lua/blob/master/sample/lua/date.lua>

[rime输入法配置英文翻译 - CSDN](<https://www.csdn.net/tags/MtTaAg3sNTAzNTI2LWJsb2cO0O0O.htm>