

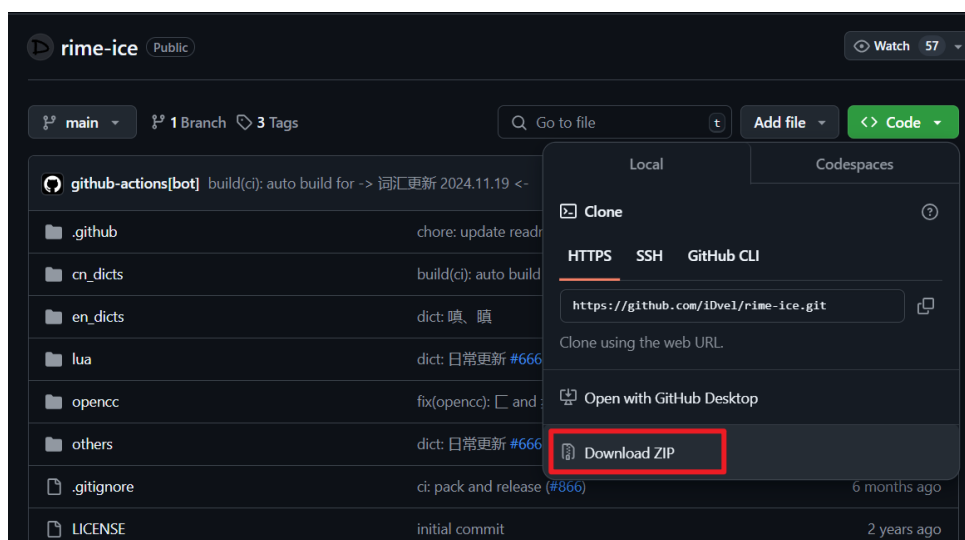
Rime 输入法官网: <https://rime.im>

雾凇拼音: <https://github.com/iDvel/rime-ice>

## Windows 端设置:

(1) 安装小狼毫输入法: 从官网下载小狼毫输入法, 直接安装即可, 安装过程中可以设置用户文档的存放位置(如果再次安装时没有显示设置位置, 可以手动打开 `C:\Program Files\Rime\weasel-0.17.4\WeaselSetup.exe` 程序进行设置), 安装完后默认就会添加到系统键盘, 如果没有的话就从系统的语言设置里面手动添加小狼毫输入法。

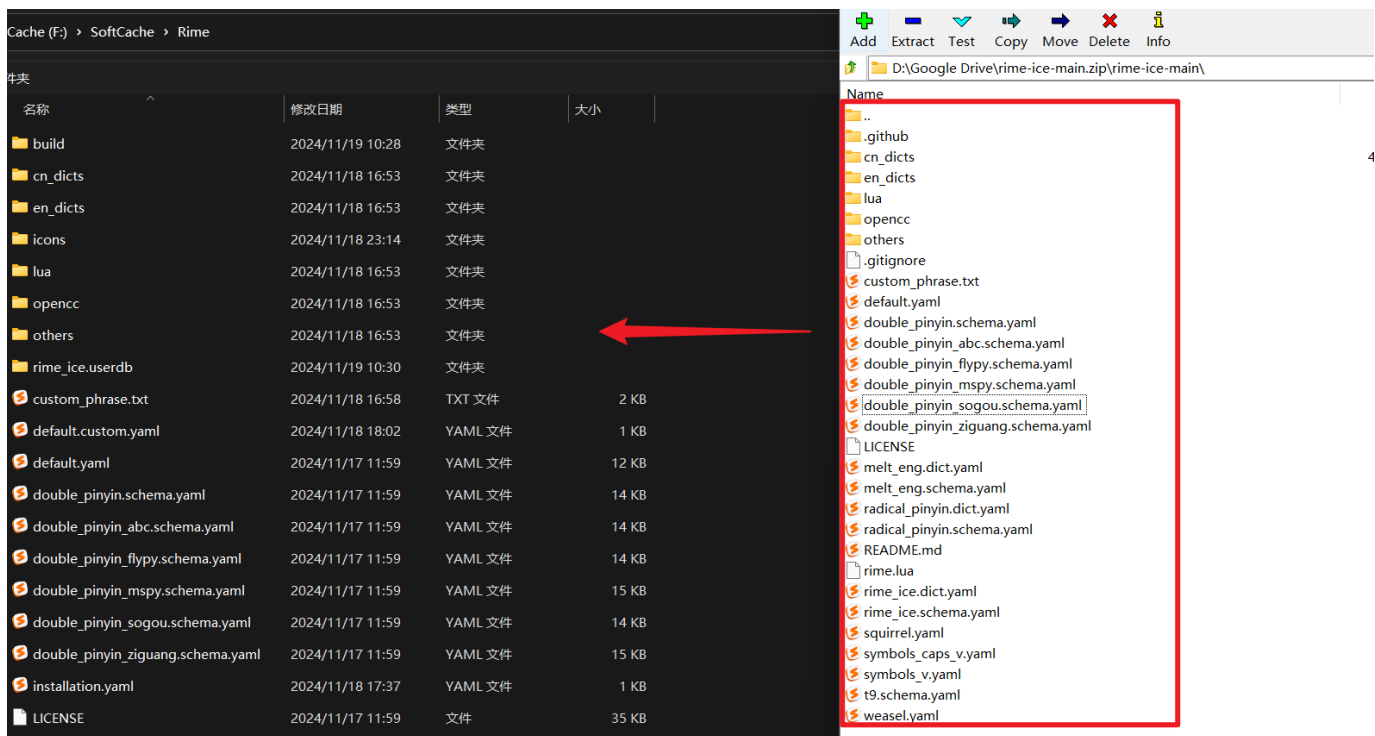
(2) 安装雾凇拼音: 直接将源码下载下来, 如下图:



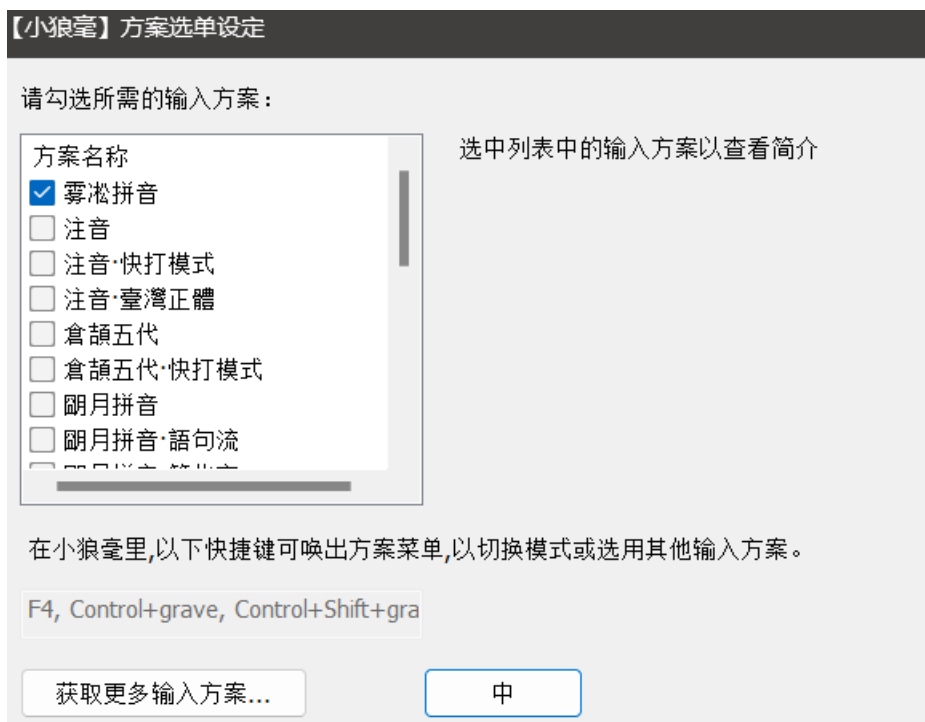
然后打开 Rime 输入法用户文件夹, 右键任务栏通知区域的 Rime 图标, 从菜单打开用户文件夹:



然后直接把下载好的源码文件夹里面的文件直接复制到 Rime 用户文件夹 (正常复制过来不会有文件冲突, 但还是建议复制前先备份下当前用户文件夹):



复制完后，右键任务栏 Rime 图标，在菜单里面选择**重新部署**，重新部署的过程大概需要十几秒钟，部署完后，右键任务栏 Rime 图标，选择**输入法设定**，就可以选择雾凇拼音了：



**(3) 更换中英输入提示图标:** Rime 默认的中英图标看着有点违和，可以自定义图标，我用的是雾凇拼音，首先编辑 `C:\Windows\SysWOW64\weasel.dll`、`C:\Windows\System32\weasel.dll` 这两个文件，用 Resources Hacker 将输入法指示图标换成自定义的，在 PE 环境下替换掉（这两个文件会影响初始载入时的图标显示）；然后将图标文件复制进用户文件夹的 icons 文件夹，编辑文件：`rime_ice.schema.yaml`

注意：编辑配置文件的时候一定要注意缩进，一段缩进是 2 个空格，2 段缩进是 4 个空格！

```
# Rime schema
# encoding: utf-8

# 方案说明
schema:
  schema_id: rime_ice
  name: 雾凇拼音
  version: "2024-02-25"
  author:
    - Dvel
  description: |
    雾凇拼音
    https://github.com/iDvel/rime-ice
  dependencies:
    - melt_eng      # 英文输入，作为次翻译器挂载到拼音方案
    - radical_pinyin # 部件拆字，反查及辅码
  icon: "icons/中.ico"
  ascii_icon: "icons/英.ico"
```


(4) 添加自定义短语：雾凇拼音的方案文件 `rime_ice.schema.yaml` 中默认设置了自定义短语文件：

```
# 自定义短语: custom_phrase.txt
custom_phrase:
  dictionary: ""
  user_dict: custom_phrase # 可以修改这里，改成自己的 txt 文件
  db_class: stabledb      # 只读数据库，无法动态调频；设为 tabledb 可以动态调频
  enable_completion: false # 补全提示
  enable_sentence: false  # 禁止造句
  initial_quality: 99     # custom_phrase 的权重应该比 pinyin 和 melt_eng 大
```

我们只需要编辑用户文件夹下的 `custom_phrase.txt`，比如我把自己的邮箱设置了快速输入：

```
main main
Amazon amazon
NASA nasa
gone gone
Go程 gocheng
code code
SOHO soho

tojohnonly@gmail.com toj
tojohnonly@hotmail.com toj
```



(5) 设置候选词数量：编辑 `default.custom.yaml` 文件，添加如下代码：

```
patch:
  "menu/page_size": 4
```

```
customization:
  distribution_code_name: Weasel
  distribution_version: 0.16.3
  generator: "Rime::SwitcherSettings"
  modified_time: "Wed Nov 20 17:33:53 2024"
  rime_version: 1.11.2
patch:
  "menu/page_size": 4
  schema_list:
    - {schema: rime_ice}
```

(6) 设置选词快捷键：编辑 `rime_ice.schema.yaml` 文件，添加如下代码：

# 从 default 继承快捷键

key\_binder:

import\_preset: default # 从 default.yaml 继承通用的

search: "" # 辅码引导符，要添加到 speller/alphabet

bindings: # 也可以再增加方案专有的快捷键

- { when: composing, accept: Tab, send: Down }
- { when: composing, accept: Left, send: Up }
- { when: composing, accept: Right, send: Down }
- { when: composing, accept: Up, send: Page\_Up }
- { when: composing, accept: Down, send: Page\_Down }

# 从 default 继承快捷键

key\_binder:

import\_preset: default # 从 default.yaml 继承通用的

search: "" # 辅码引导符，要添加到 speller/alphabet

bindings: # 也可以再增加方案专有的快捷键

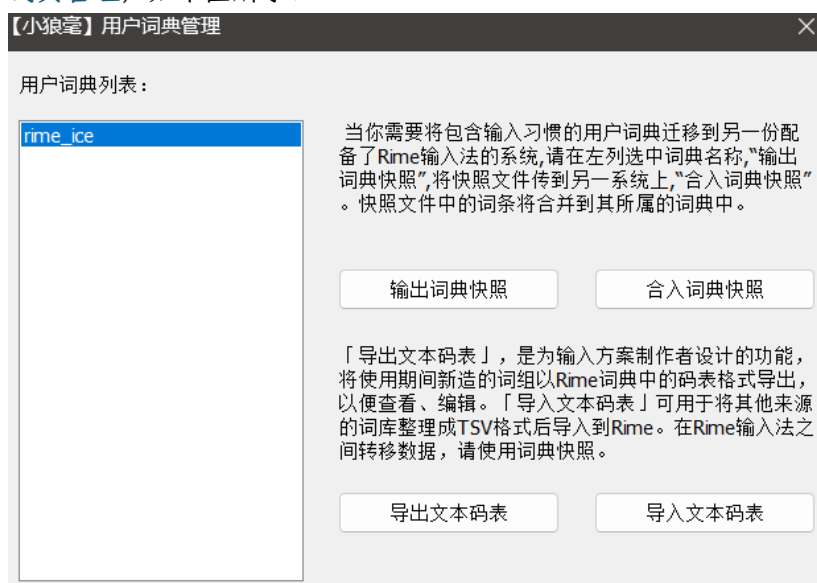
- { when: composing, accept: Tab, send: Down }
- { when: composing, accept: Left, send: Up }
- { when: composing, accept: Right, send: Down }
- { when: composing, accept: Up, send: Page\_Up }
- { when: composing, accept: Down, send: Page\_Down }

(7) 设置皮肤：编辑 `weasel.custom.yaml` 文件，添加如下代码：

```
patch:
  "preset_color_schemes/ubik2":
    author: Electric Sheep <shinzoqchiuq@outlook.com>
    back_color: 0x202020
    border_color: 0x202020
    text_color: 0xE0E0FF
    hilited_text_color: 0x4040FF
    hilited_back_color: 0x000000
    hilited_comment_text_color: 0x80D0D0
    hilited_candidate_back_color: 0x202020
    hilited_candidate_text_color: 0xFFFFFFFF
    candidate_text_color: 0x999999
    comment_text_color: 0x80D0D0
    label_color: 0x808080
    hilited_label_color: 0BBBBBB
    name: Ubik2
  "style/color_scheme": ubik2
```

(8) 管理用户词典：自己输入的学习词库可以导入或者导出，右键任务栏 Rime 图标，在菜单里面选择用户

词典管理，如下图所示：



如果自己输入错了词汇，可以用方向键定位到这个错误的词汇，按下 `Shift+Delete` 或 `Control+Delete` (MacOS 用 `Shift+Fn+Delete`) 删除即可。

(9) 去除默认的单字固定: 编辑 rime\_ice.schema.yaml 文件:

```
# Lua 配置: 置顶候选项
# 注释太长了, 请参考 pin_cand_filter.lua 开头的说明书。
pin_cand_filter:
# 格式: 编码<Tab>字词1<Space>字词2.....
# ⚙️ 以下是个人习惯, 仅供参考, 推荐打补丁用自己的习惯覆盖。
# 单编码
# - q 去 干
# - w 我 万 往
# - e 呃
# - r 让 人
# - t 他 她 它 祂
# - y 与 于
## - u 在 custom_phrase 置顶了 有 有 又 由
## - i 在 custom_phrase 置顶了 一 以 已 亦
# - o 哦
# - p 片 篇
# - a 啊
# - s 是 时 使 式
# - d 的 地 得
# - f 发 放 分
# - g 个 各
# - h 和 或
# - j 及 将 即 既 继
# - k 可
# - l 了 啦 喽 嘞
```

这里的条目都加上注释  
用自学习的排序

(10) 关闭 emoji:

```
# 开关
# reset: 默认状态。注释掉后, 切换窗口时不会重置到默认状态。
# states: 方案选单显示的名称。可以注释掉, 仍可以通过快捷键切换。
# abbrev: 默认的缩写取 states 的第一个字符, abbrev 可自定义一个字符
switches:
- name: ascii_mode
  states: [ 中, A ]
- name: ascii_punct # 中英标点
  states: [ ¥, $ ]
- name: traditionalization
  states: [ 简, 繁 ]
- name: emoji
  states: [ 🐼, 😊 ]
  reset: 0
- name: full_shape
  states: [ 半角, 全角 ]
- name: search_single_char # search.lua 的功能开关, 辅码查词时是否单字优先
  abbrev: [ 词, 单 ]
  states: [ 正常, 单字 ]
```

这里改成0, 调整emoji出现频次

**(11) 用户资料同步：**Rime 没有云同步功能，但有本地同步功能。能够将用户数据同步至本地文件夹。

我们可以借助 GoogleDrive、OneDrive 等第三方云将这个本地文件夹同步至云端，以此实现个人词典和配置方案在不同电脑间的同步和备份。

我自己用的是 GoogleDrive，新建一个同步文件夹 D:\Google Drive\Document\Rime，编辑用户文件夹的 installation.yaml 文件，添加如下代码：sync\_dir: 'D:\Google Drive\Document\Rime'，如下图：

```
distribution_code_name: Weasel
distribution_name: "小狼毫"
distribution_version: 0.16.3
install_time: "Mon Nov 18 16:52:58 2024"
installation_id: "b2-1a3"
rime_version: 1.11.2
sync_dir: 'D:\Google Drive\Document\Rime'
```

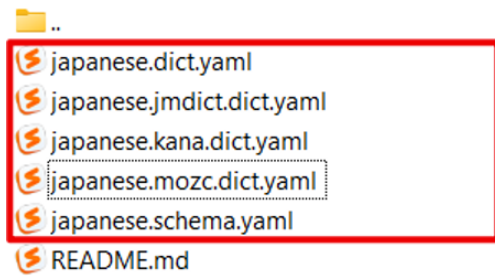
右键任务栏 Rime 图标，在菜单里面选择[用户资料同步](#)，完成后就能在刚刚新建的文件夹中找到当前机器 ID 对应的文件夹，其中的内容就是你的用户资料，包含了自学习个人词典文件和配置文件等等；然后同步到云端。

在另外一台电脑，按照相同的方式操作，将云端文件夹同步至本地，同时本机也会生成一个新的机器 ID 对应的文件夹，同步文件夹内有两个同步资料，这里讲解下 Rime 的同步逻辑：Rime 同步两个方面的资料，一为个人词典；二为个人配置；个人词典同步逻辑为双向同步；个人配置同步逻辑为单向同步。

个人词典双向同步逻辑，举例来说：甲电脑个人词典累积了词汇 ABC，乙电脑累积了词汇 DEF，那么，通过第三方云同步和 Rime 同步后，甲乙两地个人词典词汇都会同步且合并为 ABCDEF。通过第三方云同步，可以非常方便地同步两地之间的个人词典，保持相同的输入体验。

个人配置单向同步逻辑，是指 Rime 只会将配置文件，单向地从「用户文件夹 ~\Rime」同步至「同步文件夹 Rime」。换句话说，个人配置只会在「同步文件夹 Rime」里在甲乙两地被反复同步和覆盖，而不会导入配置文件，如果你需要导入异地的配置文件，可以在第三方云完成同步后，手动将配置文件导入。这样的同步逻辑是为了保持配置的一致性。因为，配置文件之于 Rime 十分重要，关系着 Rime 是否能够正常运行，也必须在修改后通过重新部署才能生效。若两地的配置不一致时或其中一地有错误时，必然产生无法预估的混乱后果。不过，个人配置定制好之后，也很少需要修改，权且当作备份。云同步个人配置更多用于异地新电脑部署时。

(12) 安装日语方案: 日语方案地址: <https://github.com/gkovacs/rime-japanese>, 下载后将以下文件复制到 Rime 用户文件夹:



Rime 发布下, 设置里面勾选日语:



然后按下 F4 键, 切换到日语:



设置输入法指示图标:

修改 `japanese.schema.yaml` 配置文件:

```
icon: "icons/あ.ico"
```

```
ascii_icon: "icons/英.ico"
```

```
5 schema:
6   schema_id: japanese
7   name: 日本語
8   version: 'v0.2'
9   author:
10    - ensigma96 <ensigma96@gmail.com>
11   dependencies:
12    - terra_pinyin.extended # for using reverse lookup
13    - stroke
14   icon: "icons/あ.ico"
15   ascii_icon: "icons/英.ico"
16
17 switches:
18   - name: ascii_mode
19     reset: 0
20     states: [ 中文, 西文 ]
```



设置自定义短语:

首先新建 custom\_phrase\_jp.txt 文件, 然后修改 `japanese.schema.yaml` 配置文件:

```
- table_translator@custom_phrase
```

```
36 engine:
37   processors:
38     - ascii_composer
39     - recognizer
40     - key_binder
41     - speller
42     - punctuator
43     - selector
44     - navigator
45     - express_editor
46   segmentors:
47     - ascii_segmentor
48     - matcher
49     - abc_segmentor
50     - affix_segmentor@putonghua_to_kanji_lookup
51     - affix_segmentor@hannom_lookup
52     - affix_segmentor@hanja_lookup
53     - punct_segmentor
54     - fallback_segmentor
55   translators:
56     - punct_translator
57     - script_translator
58     - reverse_lookup_translator
59     - script_translator@putonghua_to_kanji_lookup
60     - script_translator@hannom_lookup
61     - script_translator@hanja_lookup
62     - table_translator@custom_phrase
63   filters:
64     - simplifier
65     #- simplifier@jp_variants
```

# 自定义短语: custom\_phrase.txt

custom\_phrase:

dictionary: ""

user\_dict: custom\_phrase\_jp # 可以修改这里, 改成自己的 txt 文件

db\_class: stabledb # 只读数据库, 无法动态调频; 设为 tabledb 可以动态调频

enable\_completion: false # 补全提示

enable\_sentence: false # 禁止造句

initial\_quality: 99 # custom\_phrase 的权重

```
396 punctuator:
397   import_preset: default
398
399 # 自定义短语: custom_phrase.txt
400 custom_phrase:
401   dictionary: ""
402   user_dict: custom_phrase_jp # 可以修改这里, 改成自己的 txt 文件
403   db_class: stabledb # 只读数据库, 无法动态调频; 设为 tabledb 可以动态调频
404   enable_completion: false # 补全提示
405   enable_sentence: false # 禁止造句
406   initial_quality: 99 # custom_phrase 的权重
407
408
409 key_binder:
410   bindings:
411     - {accept: "Control+Shift+1", select: .next, when: always}
412     - {accept: "Control+Shift+n", select: japanese, when: always}
```

设置快捷键:

- { when: composing, accept: Tab, send: Down }
- { when: composing, accept: Left, send: Up }
- { when: composing, accept: Right, send: Down }
- { when: composing, accept: Up, send: Page\_Up }
- { when: composing, accept: Down, send: Page\_Down }

```
409 key_binder:
410   bindings:
411     - {accept: "Control+Shift+1", select: .next, when: always}
412     - {accept: "Control+Shift+n", select: japanese, when: always}
413     - {accept: "Control+Shift+N", select: japanese, when: always}
414     - {accept: "Control+Shift+k", select: korean, when: always}
415     - {accept: "Control+Shift+K", select: korean, when: always}
416     - {accept: "Control+Shift+v", select: vietnamese, when: always}
417     - {accept: "Control+Shift+V", select: vietnamese, when: always}
418     - {accept: "Control+Shift+j", select: double_jyutping_extra, when: always}
419     - {accept: "Control+Shift+J", select: double_jyutping_extra, when: always}
420     - {accept: "Control+Shift+m", select: td_pinyin_flypy_extra, when: always}
421     - {accept: "Control+Shift+M", select: td_pinyin_flypy_extra, when: always}
422     - {accept: "Control+Shift+l", select: international, when: always}
423     - {accept: "Control+Shift+L", select: international, when: always}
424     - {accept: "F16", select: .next, when: always}
425     - {accept: "F19", select: international, when: always}
426     # [] Pagination
427     - {accept: bracketleft, send: Page_Up , when: has_menu}
428     - {accept: bracketright, send: Page_Down , when: has_menu}
429     - { when: composing, accept: Tab, send: Down }
430     - { when: composing, accept: Left, send: Up }
431     - { when: composing, accept: Right, send: Down }
432     - { when: composing, accept: Up, send: Page_Up }
433     - { when: composing, accept: Down, send: Page_Down }
434
435 recognizer:
436   import_preset: default
```

## IOS 端设置:

(1) 安装仓输入法: 只接在 AppStore 搜索仓输入法, 图标是一只小仓鼠, 下载安装, 打开初始化后, 添加到系统键盘, 开启全键盘控制, 开启全键盘控制, 开启全键盘控制, 同步词库的时候这个设置有影响。

(2) 配置仓输入法: 我一直用的是九宫格, 所以输入方案里选择仓·九宫格 (之前是雾凇拼音·九宫格), 键盘布局里面选择中文 9 键, 同时在键盘设置里面, 可以设置键位文字大小、去掉候选栏的各种图标, 使其看着更像原生的输入法那样干净简洁。



(3) 添加自定义短语：打开 APP 界面上的文件管理，点击键盘文件 Tab，打开 Rime/t9.schema.yaml，可以看到自定义短语相关的配置：

```
62 ↵
63 #·自定义短语：custom_phrase_t9.txt↵
64 #·⚠·编码要用数字↵
65 custom_phrase:↵
66 ··dictionary:·" "↵
67 ··user_dict:·custom_phrase_t9↵
68 ··db_class:·stabledb↵
69 ··enable_completion:·false↵
70 ··enable_sentence:·false↵
71 ··initial_quality:·99↵
72 ↵
```

文件是 custom\_phrase\_t9.txt，同时快捷键位需要用九宫格对应的数字，由于没有 custom\_phrase\_t9.txt 文件，可以手动拷贝当前文件夹下面的 custom\_phrase.txt，重命名为 custom\_phrase\_t9.txt，添加如下配置：

```
61 gone▶ gone↵
62 Go程 gocheng↵
63 code▶ code↵
64 SOHO▶ soho↵
65 ↵
66 tojohnonly@gmail.com▶ 865↵
67 tojohnonly@hotmail.com▶ 865↵
68 ↵
```

这里的 865，对应的是九宫格上面 toj 键位的数字键位，然后重新部署下 Rime 就生效了（如果没有生效，就在文件管理->通用，点击拷贝键盘词库文件至应用再重新部署）。

(4) 九宫格支持英文：仓·九宫格默认不支持英文联想，需要手动开启，方案配置文件里面有说明：

```
< 方案文件管理 t9.schema.yaml 保存
1 #·Rime·schema↵
2 #·encoding:·utf-8↵
3 ↵
4 ↵
5 #·适配了仓输入法·[Hamster](https://github.com/imfuxiao/Hamster/)·九宫格布局。↵
6 #·参考于仓老师·[morse.hsiao](https://github.com/imfuxiao)·的示例↵
7 #↵
8 #·可选，支持英文：↵
9 #·1.·解开·`engine/translators`·下的注释↵
10 #·2.·将·`others/Hamster/melt_eng.custom.yaml`·里的文件拷贝至配置目录，该补丁转写了英文的拼写规则到九宫格；↵
```

需要先开启注释：

方案文件管理 t9.schema.yaml 保存

```

25 dependencies:~
26 - melt_eng~# 英文输入,
  作为次翻译器挂载到拼音方案~
27 ~
28 ~
29 engine:~
30 processors:~
31 - ascii_composer~
32 - recognizer~
33 - key_binder~
34 - speller~
35 - punctuator~
36 - selector~
37 - navigator~
38 - express_editor~
39 segmentors:~
40 - ascii_segmentor~
41 - matcher~
42 - abc_segmentor~
43 - punct_segmentor~
44 - fallback_segmentor~
45 translators:~
46 - punct_translator~
47 - script_translator~
48 - lua_translator@date_translator~# 时间、
  日期、星期~
49 - table_translator@custom_phrase~# 自定义短语 custom_phrase_t9.txt~
50 - table_translator@melt_eng~# 英文输入
51 filters:~
52 - simplifier@emoji~#
  Emoji~
53 - simplifier@traditionalize~# 简繁切换~
54 - uniquifier~#

```

然后把雾凇拼音 GIT 仓库下的 `others/Hamster/melt_eng.custom.yaml` 复制到文件管理的键盘文件 Rime 目录下 (这个可以在 IOS 自带的文件 APP 下操作)，然后重新部署即可生效。

如果找不到 `melt_eng.custom.yaml` 文件，可以手动新建，然后粘贴如下代码：

# 此补丁可以在仓输入法九宫格布局中输入英文

patch:

speller/algebra/+:

- xlit/abcdefghijklmnopqrstuvwxyz/22233344455566677778889999/

(5) 修改滑动输入判定的阈值：将 `SharedSupport/hamster.yaml` 文件复制到 Rime 文件夹下，修改滑动判定的阈值，保存后重新部署，如下图：

```

# 划动相关配置
swipe:
  # 指空格移动多个 point，光标移动一位。所以数字越小越灵敏
  spaceDragSensitivity: 15

  # 划动距离阈值，当划动的距离大于或等于此值时，才会识别为一次划动
  distanceThreshold: 80

```

(6) 修改按键 1 上滑的跳转：打开 `Rime/t9.schema.yaml`，注释 67~71 行，如下图：

```
< 方案文件管理 t9.schema.yaml 保存
```

```
65 ↵  
66 #·九宫格·1·键上的符号序列↵  
67 #punctuator:↵  
68 ··#half_shape:↵  
69 ····#"1": [1, ·"@", ·" ", ·" / ", ·" : ", ·" _ ", ·" - ",  
    " # "]↵  
70 ····#"@" : [1, ·"@", ·"! ", ·" : ", ·" \ ", ·" . ", ·" ~ "]↵  
71 ····#"#" : [0, ·"#", ·"$", ·"%", ·"^", ·"&", ·"*", ·"  
    " ( ", ·") "]"↵  
72 ↵  
73 #·输入时按·1·可以分词↵  
74 key_binder:↵  
75 ··bindings:↵  
76 ····- { when: has_menu, accept: 1, send: ·  
    apostrophe }↵  
77
```

将 `SharedSupport/hamster_swipe.yaml` 文件复制到 Rime 文件夹下，修改按键 1 的上滑输入默认直接输入 1，

保存后重新部署，如下图：

```
# 中文九宫格划动设置
chineseNineGrid:
  keys:
    - action: { chineseNineGrid: { symbol: { char: "@/.\" } } }
      swipe:
        - direction: up
          action: { character: { char: "1" } }
      callout:
        - action: { character: { char: "1" } }
        - action: { character: { char: "@" } }
        - action: { character: { char: "/" } }
        - action: { character: { char: "." } }
    - action: { chineseNineGrid: { symbol: { char: "ABC" } } }
      swipe:
        - direction: up
          action: { character: { char: "2" } }
      callout:
        - action: { character: { char: "2" } }
        - action: { character: { char: "a" } }
        - action: { character: { char: "b" } }
        - action: { character: { char: "c" } }
    - action: { chineseNineGrid: { symbol: { char: "DEF" } } }
      swipe:
        - direction: up
          action: { character: { char: "3" } }
```

(7) 用户词库同步：仓输入法首页，点击 [RIME/选择同步路径](#)，设置好同步的位置，比如我的 [iPhone/Hamster/sync](#) 然后把 PC 上的同步文件传输复制到这个文件夹内，完成后相当于这里并列多个用户资料文件夹，然后取消勾选**部署时覆盖键盘词库文件**，点击 [RIME 同步](#)、再**重新部署**，即可让同步的词库生效，在 PC 累计的词库就可以在 iPhone 上轻松打出来了，这也是我目前主要的使用方式，问题就是需要隔一段时间手动同步下。