**Schema.yaml 详解**

**开始之前**

# Rime schema

# encoding: utf-8

**描述檔**

1. name: 方案的显示名偁〔即出现于方案选单中以示人的，通常为中文〕
2. schema\_id: 方案内部名，在代码中引用此方案时以此名为正，通常由英文、数字、下划线组成
3. author: 发明人、撰写者。如果您对方案做出了修改，请保留原作者名，并将自己的名字加在后面
4. description: 请简要描述方案历史、码表来源、该方案规则等
5. dependencies: 如果本方案依赖于其它方案〔通常来说会依頼其它方案做为反查，抑或是两种或多种方案混用时〕
6. version: 版本号，在发布新版前请确保已升版本号

**示例**

schema:

name: "苍颉检字法"

schema\_id: cangjie6

author:

- "发明人 朱邦复先生、沈红莲女士"

dependencies:

- luna\_pinyin

- jyutping

- zyenpheng

description: |

第六代仓颉输入法

码表由雪斋、惜缘和crazy4u整理

version: 0.19

**开关**

通常包含以下四个：

1. ascii\_mode 是中英文转换开关。预设0为英文，1为中文
2. full\_shape 是全角符号／半角符号开关。注意，开启全角时英文字母亦为全角。0为半角，1为全角
3. extended\_charset 是字符集开关。0为CJK基本字符集，1为CJK全字符集
   * 仅table\_translator可用
4. simplification 是转化字开关。一般情况下与上同，0为不开启转化，1为转化。
   * 此选项名偁可自定义，亦可添加多套替换用字方案：

- name: zh\_cn

states: ["漢字", "汉字"]

reset: 0

或

- options: [ zh\_trad, zh\_cn, zh\_mars ]

states:

- 字型 → 汉字

- 字型 → 汉字

- 字型 → 䕼茡

reset: 0

* states: 可不写，如不写则此开关存在但不可见，可由快捷键操作
* reset: 设定默认状态〔reset可不写，此时切换窗口时不会重置到默认状态〕

**示例**

switches:

- name: ascii\_mode

states: ["中文", "西文"]

- name: full\_shape

states: ["半角", "全角"]

- name: extended\_charset

states: ["通用", "增广"]

- name: simplification

states: ["漢字", "汉字"]

**引擎**

* 以下**加粗**项为可细配者，*斜体*者为不常用者

引擎分四组：

**一、processors**

* 这批组件处理各类按键消息

1. ascii\_composer 处理西文模式及中西文切
2. **recognizer** 与matcher搭配，处理符合特定规则的输入码，如网址、反查等tags
3. **key\_binder** 在特定条件下将按键绑定到其他按键，如重定义逗号、句号为候选翻页、开关快捷键等
4. **speller** 拼写处理器，接受字符按键，编辑输入
5. **punctuator** 句读处理器，将单个字符按键直接映像为标点符号或文字
6. selector 选字处理器，处理数字选字键〔可以换成别的哦〕、上、下候选定位、换页
7. navigator 处理输入栏内的光标移动
8. express\_editor 编辑器，处理空格、回车上屏、回退键
9. *fluency\_editor* 句式编辑器，用于以空格断词、回车上屏的【注音】、【语句流】等输入方案，替换express\_editor
10. *chord\_composer* 和弦作曲家或曰并击处理器，用于【宫保拼音】等多键并击的输入方案

**二、segmentors**

* 这批组件识别不同内容类型，将输入码分段并加上tag

1. ascii\_segmentor 标识西文段落〔譬如在西文模式下〕字母直接上屛
2. matcher 配合recognizer标识符合特定规则的段落，如网址、反查等，加上特定tag
3. **abc\_segmentor** 标识常规的文字段落，加上abc这个tag
4. punct\_segmentor 标识句读段落〔键入标点符号用〕加上punct这个tag
5. fallback\_segmentor 标识其他未标识段落
6. **affix\_segmentor** 用户自定义tag
   * 此项可加载多个实例，后接@+tag名

**三、translators**

* 这批组件翻译特定类型的编码段为一组候选文字

1. echo\_translator 没有其他候选字时，回显输入码〔输入码可以Shift+Enter上屛〕
2. punct\_translator 配合punct\_segmentor转换标点符号
3. **table\_translator** 码表翻译器，用于仓颉、五笔等基于码表的输入方案

- 此项可加载多个实例，后接@+翻译器名〔如：cangjie、wubi等〕

1. **script\_translator** 脚本翻译器，用于拼音、粤拼等基于音节表的输入方案

- 此项可加载多个实例，后接@+翻译器名〔如：pinyin、jyutping等〕

1. *reverse\_lookup\_translator* 反查翻译器，用另一种编码方案查码

**四、filters**

* 这批组件过滤翻译的结果

1. **simplifier** 用字转换
2. uniquifier 过滤重复的候选字，有可能来自**simplifier**
3. **reverse\_lookup\_filter** 反查滤镜，以更灵活的方式反查，Rime1.0后替代*reverse\_lookup\_translator*
   * 此项可加载多个实例，后接@+滤镜名〔如：pinyin\_lookup、jyutping\_lookup等〕

**示例**

cangjie6.schema.yaml

engine:

processors:

- ascii\_composer

- recognizer

- key\_binder

- speller

- punctuator

- selector

- navigator

- express\_editor

segmentors:

- ascii\_segmentor

- matcher

- affix\_segmentor@pinyin

- affix\_segmentor@jyutping

- affix\_segmentor@pinyin\_lookup

- affix\_segmentor@jyutping\_lookup

- affix\_segmentor@reverse\_lookup

- abc\_segmentor

- punct\_segmentor

- fallback\_segmentor

translators:

- punct\_translator

- table\_translator

- script\_translator@pinyin

- script\_translator@jyutping

- script\_translator@pinyin\_lookup

- script\_translator@jyutping\_lookup

filters:

- simplifier@zh\_simp

- uniquifier

- reverse\_lookup\_filter@middle\_chinese

- reverse\_lookup\_filter@pinyin\_reverse\_lookup

- reverse\_lookup\_filter@jyutping\_reverse\_lookup

**细项配置**

* 凡comment\_format、preedit\_format、speller/algebra所用之正则表达式，请参阅[「Perl正则表达式」](http://www.boost.org/doc/libs/1_49_0/libs/regex/doc/html/boost_regex/syntax/perl_syntax.html)

**引擎中所举之加粗者均可在下方详细描述，格式为：**

name:

branches: configurations

或

name:

branches:

- configurations

**一、speller**

1. alphabet: 定义本方案输入键
2. initials: 定义仅作始码之键
3. finals: 定义仅作末码之键
4. delimiter: 上屛时的音节间分音符
5. algebra: 拼写运算规则，由之算出的拼写汇入prism中
6. max\_code\_length: 形码最大码长，超过则顶字上屛〔number〕
7. auto\_select: 自动上屛〔true或false〕
8. use\_space: 以空格作输入码〔true或false〕
   * speller的演算包含：

xform --改写〔不保留原形〕

derive --衍生〔保留原形〕

abbrev --简拼〔出字优先级较上两组更低〕

fuzz --畧拼〔此种简拼仅组词，不出单字〕

xlit --变换〔适合大量一对一变换〕

erase --删除

**示例**

luna\_pinyin.schema.yaml

speller:

alphabet: zyxwvutsrqponmlkjihgfedcba

delimiter: " '"

algebra:

- erase/^xx$/

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

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

- 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/

**二、segmentor**

* segmentor配合recognizer标记出tag。这里会用到affix\_segmentor和abc\_translator
* tag用在translator、reverse\_lookup\_filter、simplifier中用以标定各自作用范围
* 如果不需要用到extra\_tags则不需要单独配置segmentor

1. tag: 设定其tag
2. prefix: 设定其前缀标识，可不塡，不塡则无前缀
3. suffix: 设定其尾缀标识，可不塡，不塡则无尾缀
4. tips: 设定其输入前提示符，可不塡，不塡则无提示符
5. closing\_tips: 设定其结束输入提示符，可不塡，不塡则无提示符
6. extra\_tags: 为此segmentor所标记的段落插上其它tag

**当affix\_segmentor和translator重名时，两者可并在一处配置，此处1-5条对应下面16-20条。abc\_segmentor仅可设extra\_tags**

**示例**

cangjie6.schema.yaml

reverse\_lookup:

tag: reverse\_lookup

prefix: "`"

suffix: ";"

tips: "【反查】"

closing\_tips: "【苍颉】"

extra\_tags:

- pinyin\_lookup

- jyutping\_lookup

**三、translator**

* 每个方案有一个主translator，在引擎列表中不以@+翻译器名定义，在细项配置时直接以translator:命名。以下加粗项为可在主translator中定义之项，其它可在副〔以@+翻译器名命名〕translator中定义

1. **enable\_charset\_filter:** 是否开启字符集过滤〔仅table\_translator有效〕
2. **enable\_encoder:** 是否开启自动造词〔仅table\_translator有效〕
3. **encode\_commit\_history:** 是否对已上屛词自动成词〔仅table\_translator有效〕
4. **max\_phrase\_length:** 最大自动成词词长〔仅table\_translator有效〕
5. **enable\_completion:** 提前显示尚未输入完整码的字〔仅table\_translator有效〕
6. **sentence\_over\_completion:** 在无全码对应字而仅有逐键提示时也开启智能组句〔仅table\_translator有效〕
7. **strict\_spelling:** 配合speller中的fuzz规则，仅以畧拼码组词〔仅table\_translator有效〕
8. **disable\_user\_dict\_for\_patterns:** 禁止某些编码录入用户词典
9. **enable\_sentence:** 是否开启自动造句
10. **enable\_user\_dict:** 是否开启用户词典〔用户词典记录动态字词频、用户词〕
    * 以上选塡true或false
11. **dictionary:** 翻译器将调取此字典文件
12. **prism:** 设定由此主翻译器的speller生成的棱镜文件名，或此副编译程序调用的棱镜名
13. **user\_dict:** 设定用户词典名
14. **db\_class:** 设定用户词典类型，可设tabledb〔文本〕或userdb〔二进制〕
15. **preedit\_format:** 上屛码自定义
16. **comment\_format:** 提示码自定义
17. **spelling\_hints:** 设定多少字以内候选标注完整带调拼音〔仅script\_translator有效〕
18. **initial\_quality:** 设定此翻译器出字优先级
19. tag: 设定此翻译器针对的tag。可不塡，不塡则仅针对abc
20. prefix: 设定此翻译器的前缀标识，可不塡，不塡则无前缀
21. suffix: 设定此翻译器的尾缀标识，可不塡，不塡则无尾缀
22. tips: 设定此翻译器的输入前提示符，可不塡，不塡则无提示符
23. closing\_tips: 设定此翻译器的结束输入提示符，可不塡，不塡则无提示符

**示例**

cangjie6.schema.yaml 苍颉主翻译器

translator:

dictionary: cangjie6

enable\_charset\_filter: true

enable\_sentence: true

enable\_encoder: true

encode\_commit\_history: true

max\_phrase\_length: 5

preedit\_format:

- xform/^([a-z ]*)$/$1｜\U$1\E/*

*- xform/(?<=[a-z])\s(?=[a-z])//*

*- "xlit|ABCDEFGHIJKLMNOPQRSTUVWXYZ|日月金木水火土竹戈十大中一弓人心手口尸廿山女田止卜片|"*

*comment\_format:*

*- "xlit|abcdefghijklmnopqrstuvwxyz~|日月金木水火土竹戈十大中一弓人心手口尸廿山女田止卜片・|"*

*disable\_user\_dict\_for\_patterns:*

*- "^z.*$"

initial\_quality: 0.75

cangjie6.schema.yaml 拼音副翻译器

pinyin:

tag: pinyin

dictionary: luna\_pinyin

enable\_charset\_filter: true

prefix: 'P' #须配合recognizer

suffix: ';' #须配合recognizer

preedit\_format:

- "xform/([nl])v/$1ü/"

- "xform/([nl])ue/$1üe/"

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

tips: "【汉拼】"

closing\_tips: "【苍颉】"

pinyin\_simp.schema.yaml 拼音・简化字主翻译器

translator:

dictionary: luna\_pinyin

prism: luna\_pinyin\_simp

preedit\_format:

- xform/([nl])v/$1ü/

- xform/([nl])ue/$1üe/

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

luna\_pinyin.schema.yaml 朙月拼音用户短语

custom\_phrase: #这是一个table\_translator

dictionary: ""

user\_dict: custom\_phrase

db\_class: tabledb

enable\_sentence: false

enable\_completion: false

initial\_quality: 1

**四、reverse\_lookup\_filter**

* 此滤镜须挂在translator上，不影响该translator工作

1. tags: 设定其作用范围
2. overwrite\_comment: 是否覆盖其他提示
3. dictionary: 反查所得提示码之码表
4. comment\_format: 自定义提示码格式

**示例**

cangjie6.schema.yaml

pinyin\_reverse\_lookup: #该反查滤镜名

tags: [ pinyin\_lookup ] #挂在这个tag所对应的翻译器上

overwrite\_comment: true

dictionary: cangjie6 #反查所得为苍颉码

comment\_format:

- "xform/$/〕/"

- "xform/^/〔/"

- "xlit|abcdefghijklmnopqrstuvwxyz |日月金木水火土竹戈十大中一弓人心手口尸廿山女田止卜片、|"

**五、simplifier**

1. option\_name: 对应swiches中设定的切换项名
2. opencc\_config: 用字转换定义文件
3. tags: 设定转换范围
4. tips: 设定是否提示转换前的字，可塡none〔或不塡〕、char〔仅对单字有效〕、all
5. *excluded\_types:* 取消特定范围〔一般为*reverse\_lookup\_translator*〕转化用字

**示例**

modified from luna\_pinyin\_kunki.schema

zh\_tw:

option\_name: zh\_tw

opencc\_config: zht2zhtw\_p.ini

tags: [ abc ] #abc对应abc\_segmentor

tips: none

***六、chord\_composer***

* 并击把键盘分两半，相当于两块键盘。两边同时击键，系统默认在其中一半上按的键先于另一半，由此得出上屛码

1. alphabet: 字母表，包含用于并击的按键。击键虽有先后，形成并击时，一律以字母表顺序排列
2. algebra: 拼写运算规则，将一组并击编码转换为拼音音节
3. output\_format: 并击完成后套用的式样，追加隔音符号
4. prompt\_format: 并击过程中套用的式样，加方括号

**示例**

combo\_pinyin.schema.yaml

chord\_composer:

# 字母表，包含用于并击的按键

# 击键虽有先后，形成并击时，一律以字母表顺序排列

alphabet: "swxdecfrvgtbnjum ki,lo."

# 拼写运算规则，将一组并击编码转换为拼音音节

algebra:

# 先将物理按键字符对应到宫保拼音键位中的拼音字母

- 'xlit|swxdecfrvgtbnjum ki,lo.|sczhlfgdbktpRiuVaNIUeoE|'

# 以下根据宫保拼音的键位分别变换声母、韵母部分

# 组合声母

- xform/^zf/zh/

- xform/^cl/ch/

- xform/^fb/m/

- xform/^ld/n/

- xform/^hg/r/

……

# 声母独用时补足隠含的韵母

- xform/^([bpf])$/$1u/

- xform/^([mdtnlgkh])$/$1e/

- xform/^([mdtnlgkh])$/$1e/

- xform/^([zcsr]h?)$/$1i/

# 并击完成后套用的式样，追加隔音符号

output\_format:

- "xform/^([a-z]+)$/$1'/"

# 并击过程中套用的式样，加方括号

prompt\_format:

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

**七、其它**

* 包括recognizer、key\_binder、punctuator。**标点**、**快捷键**、**二三选重**、**特殊字符**等均于此设置

1. **import\_preset:** 由外部统一文件导入
2. recognizer: 下设patterns: 配合segmentor的prefix和suffix完成段落划分、tag分配
3. key\_binder: 下设bindings: 设置功能性快捷键
4. punctuator: 下设full\_shape:和half\_shape:分别控制全角模式下的符号和半角模式下的符号，另有use\_space:空格顶字

**示例**

modified from cangjie6.schema.yaml

key\_binder:

import\_preset: default

bindings:

- {accept: semicolon, send: 2, when: has\_menu} #分号选第二重码

- {accept: apostrophe, send: 3, when: has\_menu} #引号选第三重码

- {accept: "Control+1", select: .next, when: always}

- {accept: "Control+2", toggle: full\_shape, when: always}

- {accept: "Control+3", toggle: simplification, when: always}

- {accept: "Control+4", toggle: extended\_charset, when: always}

punctuator:

import\_preset: symbols

half\_shape:

"'": {pair: ["「", "」"]} #第一次按是「，第二次是」

"(": ["〔", "［"] #弹出选单

.: {commit: "。"} #无选单，直接上屛。优先级最高

recognizer:

import\_preset: default

patterns:

email: "^[a-z][-\_.0-9a-z]\*@.\*$"

url: "^(www[.]|https?:|ftp:|mailto:).\*$"

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

pinyin\_lookup: "`P[a-z]\*;?$"

jyutping\_lookup: "`J[a-z]\*;?$"

pinyin: "(?<!`)P[a-z']\*;?$"

jyutping: "(?<!`)J[a-z']\*;?$"

punct: "/[a-z]\*$" #配合symbols.yaml中的特殊字符输入

**其它**

* Rime还为每个方案提供选单和一定的外观订制能力
* 通常情况下menu在default.yaml中定义〔或用户修改文件default.custom.yaml〕，style在squirrel.yaml或weasel.yaml〔或用户修改文件squirrel.custom.yaml或weasel.custom.yaml〕

menu:

alternative\_select\_keys: ASDFGHJKL #如编码字符占用数字键则须另设选字键

page\_size: 5 #选单每䈎显示个数

style:

font\_face: "HanaMinA, HanaMinB" #字体〔小狼毫得且仅得设一个字体；鼠须管得设多个字体，后面的字体自动补前面字体不含的字〕

font\_point: 15 #字号

horizontal: false #横／直排

line\_spacing: 1 #行距

inline\_preedit: true #输入码内嵌

**Dict.yaml 详解**

**开始之前**

# Rime dict

# encoding: utf-8

〔你还可以在这注释字典来源、变动记䤸等〕

**描述檔**

1. name: 内部字典名，也即schema所引用的字典名，确保与文件名相一致
2. version: 如果发布，请确保每次改动升版本号

**示例**

name: "cangjie6.extended"

version: "0.1"

**配置**

1. sort: 字典**初始**排序，可选original或by\_weight
2. use\_preset\_vocabulary: 是否引入「八股文」〔含字词频、词库〕
3. max\_phrase\_length: 配合use\_preset\_vocabulary:，设定导入词条最大词长
4. min\_phrase\_weight: 配合use\_preset\_vocabulary:，设定导入词条最小词频
5. columns: 定义码表以Tab分隔出的各列，可设text【文本】、code【码】、weight【权重】、stem【造词码】
6. import\_tables: 加载其它外部码表
7. encoder: 形码造词规则
   1. exclude\_patterns:
   2. rules: 可用length\_equal:和length\_in\_range:定义。大写字母表示字序，小写字母表示其所跟随的大写字母所以表的字中的编码序
   3. tail\_anchor: 造词码包含结构分割符〔仅用于仓颉〕
   4. exclude\_patterns 取消某编码的造词资格

**示例**

cangjie6.extended.dict.yaml

sort: by\_weight

use\_preset\_vocabulary: false

import\_tables:

- cangjie6 #单字码表由cangjie6.dict.yaml导入

columns: #此字典为纯词典，无单字编码，仅有字和词频

- text #字／词

- weight #字／词频

encoder:

exclude\_patterns:

- '^z.\*$'

rules:

- length\_equal: 2 #对于二字词

formula: "AaAzBaBbBz" #取第一前缀后缀、第二前缀次后缀

- length\_equal: 3 #对于三字词

formula: "AaAzBaYzZz" #取第一前缀后缀、第二前缀后缀、第三字后缀

- length\_in\_range: [4, 5] #对于四至五字词

formula: "AaBzCaYzZz" #取第一字前缀，第二字后缀、第三字前缀、倒数第二字后缀、最后一字后缀

tail\_anchor: "'"

**码表**

* 以Tab分隔各列，各列依columns:定义排列。

**示例**

cangjie6.dict.yaml

columns:

- text #第一列字／词

- code #第二列码

- weight #第三列字／词频

- stem #第四列造词码

cangjie6.dict.yaml

个 owjr 246268 ow'jr

看 hqbu 245668

中 l 243881

呢 rsp 242970

来 doo 235101

吗 rsqf 221092

为 bhnf 211340

会 owfa 209844

她 vpd 204725

与 xyc 203975

给 vfor 193007

等 hgdi 183340

这 yymr 181787

用 bq 168934 b'q

雪斋  
09-Nov-2013