

1. RIME 输入法简介

弓辰，号佛振，出身地是河南郑州以北十里弓家寨，在北平做码农。佛振是[\[Rime / 中州韵\]](#)输入法引擎的开发者，它不仅仅是一个输入法，也是一个输入法算法框架，这一套算法支持了拼音、双拼、注音、五笔、仓颉等所有音码和形码输入法。Rime 还支持了许多种方言拼音，如吴语、粤语，甚至中古汉语。「IT 学生界大神」BYVoid 曾在其博客[撰文怒赞](#)「Rime / 中州韵」，使得这个原本默默无闻、志在用十年磨砺的项目在一个小圈子里迅速传开。佛振四年前加入百度 PC 输入法开发团队，但他并没有放弃自己的「业余作品」，而是将其发扬光大，针对 Windows、Linux、Mac 三大平台提供了不同的发行版，Windows 版「小狼毫」，Mac 版「鼠须管」，Linux 版「中州韵」。

Rime 是一款输入法，你也理解它是[输入法内核](#)。同时针对不同的平台它又有不同的前端，这些前端又有自己的名字。

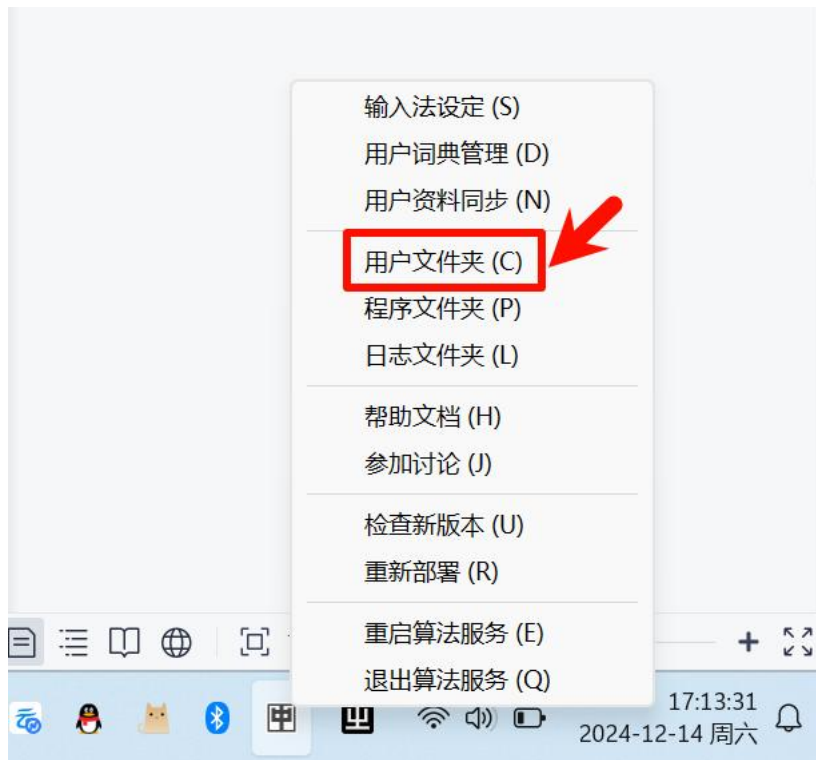
- Windows: [weasel](#)，中文也叫“小狼毫”。
- macOS: [squirrel](#)，中文也叫“鼠须管”。
- Linux: [ibus-rime](#)，中文也叫“中州韵”。
- Android: [link](#)，中文也叫“同文输入法”。

2. 首次安装配置

下载 Rime 输入法安装文件后直接默认安装，不需要修改任何信息，包括用户文件位置也不需要修改。

① 安装 Rime


② 通过任务栏图标 → 右键 → 用户文件夹，打开用户文件夹，通常默认为 C:\Users\Administrator\AppData\Roaming\Rime





初次安装完成后，用户文件夹目录下正常只有几个文件，如下

> 周星 > AppData > Roaming > Rime >

在 Rime 中搜索







排序

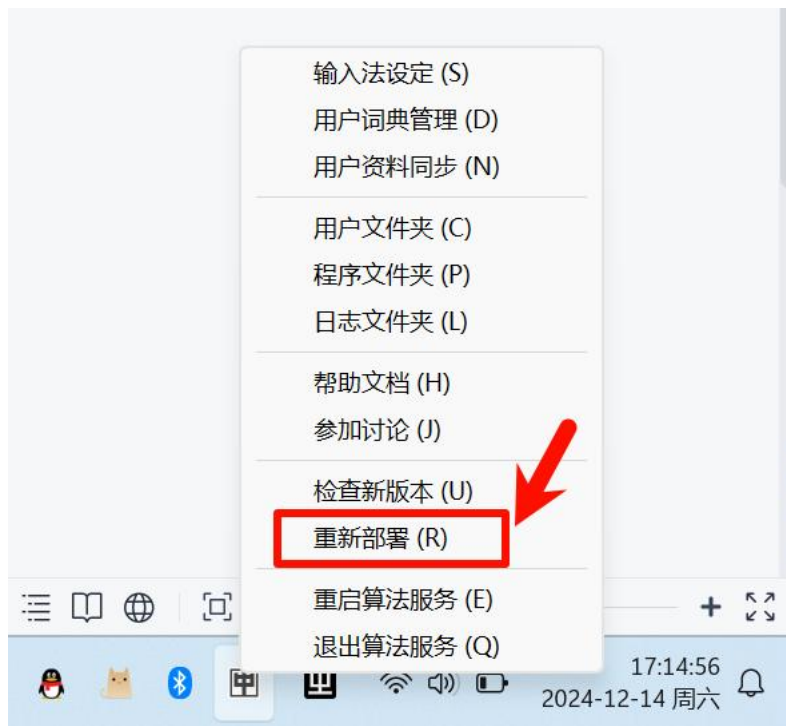
查看

名称	修改日期	类型	大小
build	2024-12-14 周六 21:50	文件夹	
luna_pinyin.userdb	2024-12-14 周六 21:50	文件夹	
default.custom.yaml	2024-12-14 周六 21:50	Yaml 源文件	0 KB
installation.yaml	2024-12-14 周六 21:50	Yaml 源文件	1 KB
user.yaml	2024-12-14 周六 21:50	Yaml 源文件	1 KB
weasel.custom.yaml	2024-12-14 周六 21:50	Yaml 源文件	0 KB

然后将**输入方案及配置文件共 9 个文件**（如下图）拷贝至**用户文件夹目录**覆盖掉原来的文件

> ... ASUS (D:) > NAS > sw > Rime > Sync > Backup					在 Backup 中搜索
📁 📄 🗑️ ↕️ 排序 ▾ ≡ 查看 ▾ ...					🔍 预览
名称	状态	修改日期	类型	大小	
default.custom.yaml	✓	2024-12-17 周二 14:59	Yaml 源文件	1 KB	
pinyin.custom.yaml	✓	2024-12-15 周日 11:23	Yaml 源文件	1 KB	
pinyin.dict.yaml	✓	2024-07-11 周四 07:28	Yaml 源文件	1,237 KB	
pinyin.schema.yaml	✓	2024-12-15 周日 10:05	Yaml 源文件	2 KB	
weasel.custom.yaml	✓	2024-12-15 周日 12:16	Yaml 源文件	6 KB	
wubi86.custom.yaml	✓	2024-12-15 周日 11:24	Yaml 源文件	1 KB	
wubi86.dict.yaml	✓	2024-12-17 周二 15:02	Yaml 源文件	1,483 KB	
wubi86.schema.yaml	✓	2024-12-15 周日 10:07	Yaml 源文件	3 KB	
wubi86_user.dict.yaml	✓	2024-12-15 周日 11:36	Yaml 源文件	19 KB	

③在任务栏右键点击 Rime 输入法图标进行重新部署即可使用



3. Rime 文件说明

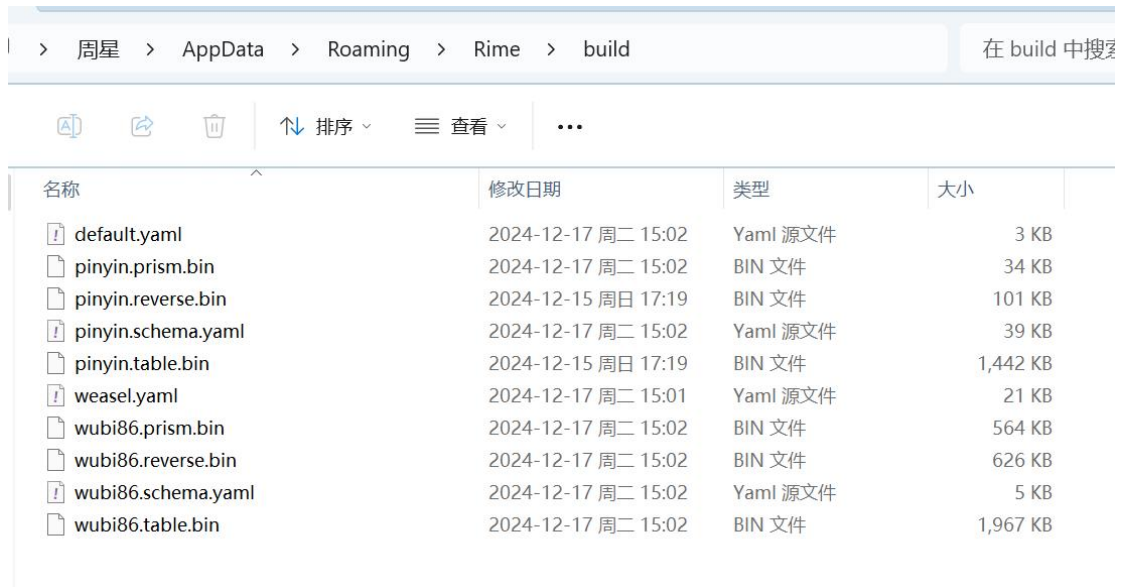
正常来说，用户目录下应该有如下文件

> 周星 > AppData > Roaming > Rime > <div>在 Rime 中搜索</div>			
<div> <div> 排序 查看 ... </div> </div>			
名称	修改日期	类型	大小
build	2024-12-17 周二 15:02	文件夹	
pinyin.userdb	2024-12-17 周二 15:21	文件夹	
wubi86.userdb	2024-12-17 周二 15:22	文件夹	
default.custom.yaml	2024-12-17 周二 14:59	Yaml 源文件	1 KB
installation.yaml	2024-12-14 周六 17:04	Yaml 源文件	1 KB
pinyin.custom.yaml	2024-12-15 周日 11:23	Yaml 源文件	1 KB
pinyin.dict.yaml	2024-07-11 周四 07:28	Yaml 源文件	1,237 KB
pinyin.schema.yaml	2024-12-15 周日 10:05	Yaml 源文件	2 KB
user.yaml	2024-12-17 周二 15:18	Yaml 源文件	1 KB
weasel.custom.yaml	2024-12-15 周日 12:16	Yaml 源文件	6 KB
wubi86.custom.yaml	2024-12-15 周日 11:24	Yaml 源文件	1 KB
wubi86.dict.yaml	2024-12-17 周二 15:02	Yaml 源文件	1,483 KB
wubi86.schema.yaml	2024-12-15 周日 10:07	Yaml 源文件	3 KB
wubi86_user.dict.yaml	2024-12-15 周日 11:36	Yaml 源文件	19 KB

上述文件简略说明如下

文件名或目录名	说明
build	编译的二进制文件
pinyin.userdb	用户数据（拼音输入法）
wubi86.userdb	用户数据（五笔输入法）
user.yaml	用户信息
installation.yaml	安装文件（配置同步参数）
default.custom.yaml	配置文件（所有输入法的默认配置）
weasel.custom.yaml	皮肤配色文件
pinyin.custom.yaml	配置文件（拼音输入法）
pinyin.dict.yaml	词库文件（拼音输入法）
pinyin.schema.yaml	输入法方案（拼音输入法）
wubi86.custom.yaml	配置文件（五笔输入法）
wubi86.dict.yaml	词库文件（五笔输入法）
wubi86.schema.yaml	输入法方案（五笔输入法）
wubi86_user.dict.yaml	用户自定义词库文件（五笔输入法）

如果打开 **build** 目录，会发现文件如下（我已删除其他不用的输入法方案）



注意到其中有 4 个重要文件，分别是

default.yaml

weasel.yaml

pinyin.schema.yaml

wubi86.schema.yaml

这 4 个文件每次更新程序或重新安装软件时会被覆写，所以我们不直接修改 build 目录下的文件，而是修改用户目录下的文件，通过重新部署生效。

修改的文件	重新部署后 build 目录下被更新的文件
default.custom.yaml	default.yaml
weasel.custom.yaml	weasel.yaml
pinyin.custom.yaml	pinyin.schema.yaml
pinyin.schema.yaml	
wubi86.custom.yaml	wubi86.schema.yaml
wubi86.schema.yaml	

网上的很多教程教用户直接在用户文件夹的 build 目录下修改 default.yaml，输入方案.schema.yaml（这里 输入方案 只是某种具体方案的代称），weasel.yaml 等文件，这是不正确的方式，因为 Rime 升级时，这些文件会被覆盖。正确的方式是，在用户文件夹下建立 default.custom.yaml 和 {输入方

案}.custom.yaml ({输入方案} 代表某种具体输入方案), weasel.custom.yaml 文件, 这些文件相当于是补丁, 部署以后会更新到默认的配置里面。

数据文件和目录在各版本中可能会有不同。

主要文件

default.yaml, 全局设定

weasel.yaml, 发行版设定

installation.yaml, 安装信息(可配置同步目录)

user.yaml, 用户状态信息

***.schema.yaml, 输入方案**

***.custom.yaml, 用户对 Rime 全局配置、各输入方案配置、自制输入方案等定制文件**

*.dict.yaml, 输入方案配套的词典源文件

用户配置同步目录及词典文件

*.userdb, Rime 记录用户输入习惯的目录文件

UUID/, 用户配置同步目录

UUID/*.userdb.txt, Rime 生成的用以同步的用户词典

Rime 编译生成的二进制文件

build/*

*.prism.bin, 棱镜文件

*.reverse.bin, 反查词典

*.table.bin, 固态词典

default.yaml→全局设定→在 build 目录下, 更新软件时会被覆盖

weasel.yaml→发行版设定→在 build 目录下，更新软件时会被覆盖

〈方案标识〉.schema.yaml→预设输入方案

〈词典名〉.userdb→记录用户输入习惯的文件

default.custom.yaml→用于设置输入法方案、切换输入法快捷键、中英文切换、翻页等等。

weasel.custom.yaml→用户对发行版设定的定制信息，用于设置托盘图标、指定软件默认英文输入、候选词横竖排列、界面布局、配色方案等等。

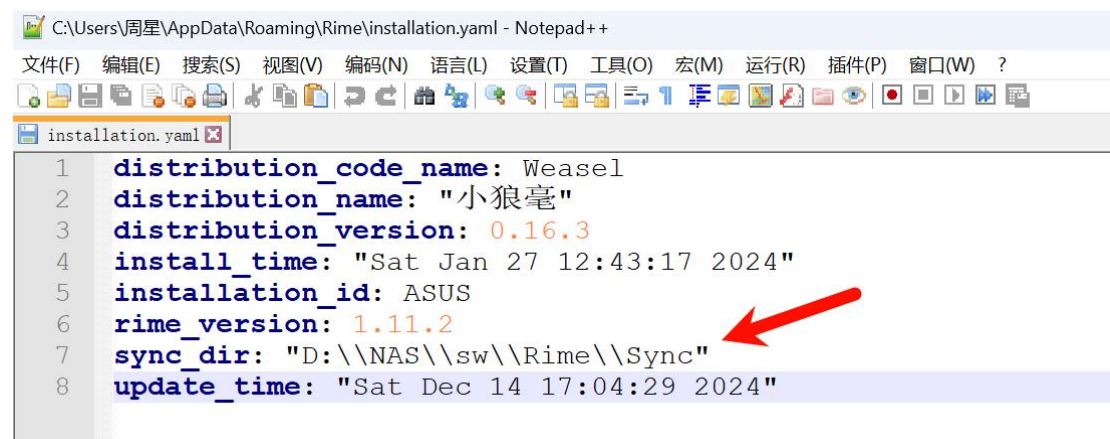
〈方案标识〉.custom.yaml→用户对预设输入方案的定制信息，用于设置具体的输入方案。

所有对如上三个 custom 配置文档的修改，最终都会以「补丁」的方式映射汇入到 default.yaml、weasel.yaml、和 〈方案标识〉.schema.yaml 三个文本文档中，经由重新部署而生效。

4. 同步配置

打开用户文件夹中的 installation.yaml 这个文件，修改下面这句话即可。注意修改完后需要重新部署。

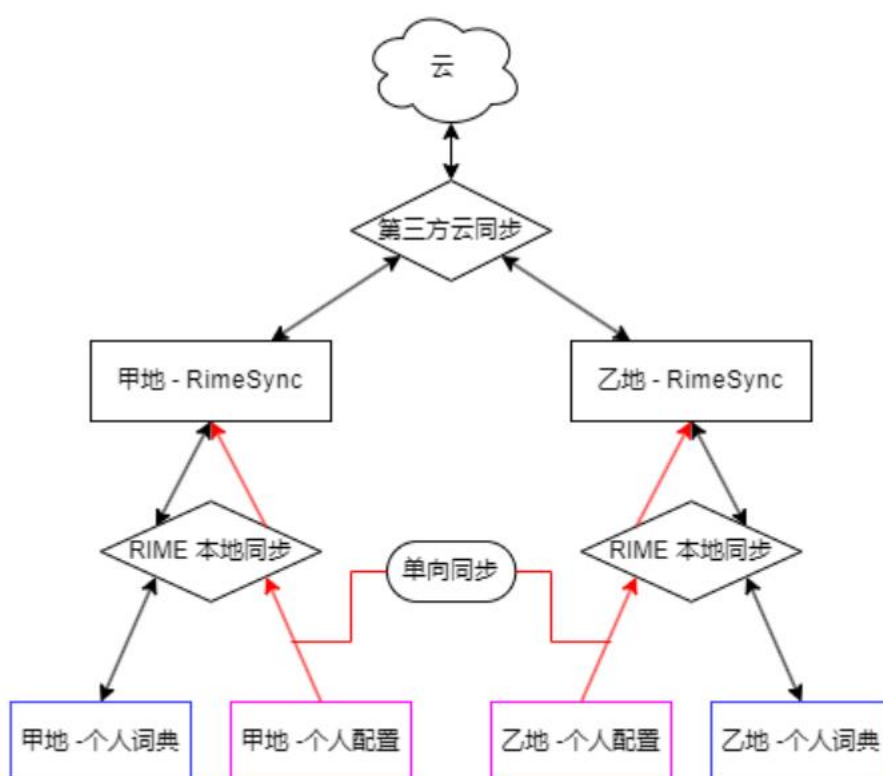
sync_dir: "D:\\NAS\\sw\\Rime\\Sync"



```
1 distribution_code_name: Weasel
2 distribution_name: "小狼毫"
3 distribution_version: 0.16.3
4 install_time: "Sat Jan 27 12:43:17 2024"
5 installation_id: ASUS
6 rime_version: 1.11.2
7 sync_dir: "D:\\NAS\\sw\\Rime\\Sync"
8 update_time: "Sat Dec 14 17:04:29 2024"
```

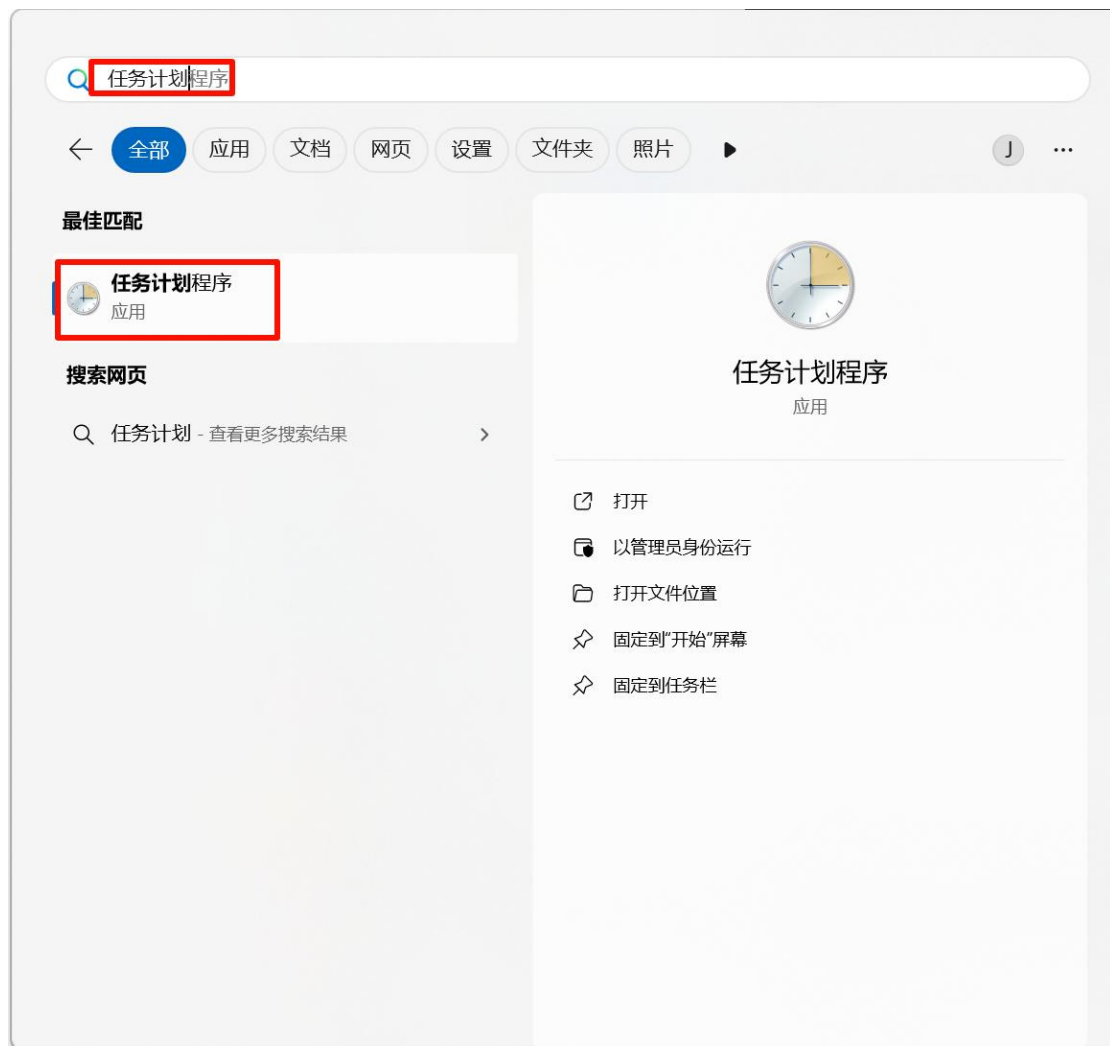
- RIME 同步两个方面的资料，一为**个人词典**；二为**个人配置**。
- 个人词典同步逻辑为双向同步；个人配置同步逻辑为单向同步。
- 个人词典双向同步逻辑，举例来说：甲电脑个人词典累积了词汇 ABC，乙电脑累积了词汇 DEF，那么，通过第三方云同步和 Rime 同步后，甲乙两地个人词典词汇都会同步且合并为 ABCDEF。通过第三方云同步，可以非常方便地同步两地之间的个人词典，保持相同的输入体验。

- 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 RimeSync」。换句话说,个人配置只会在「同步文件夹 RimeSync」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。

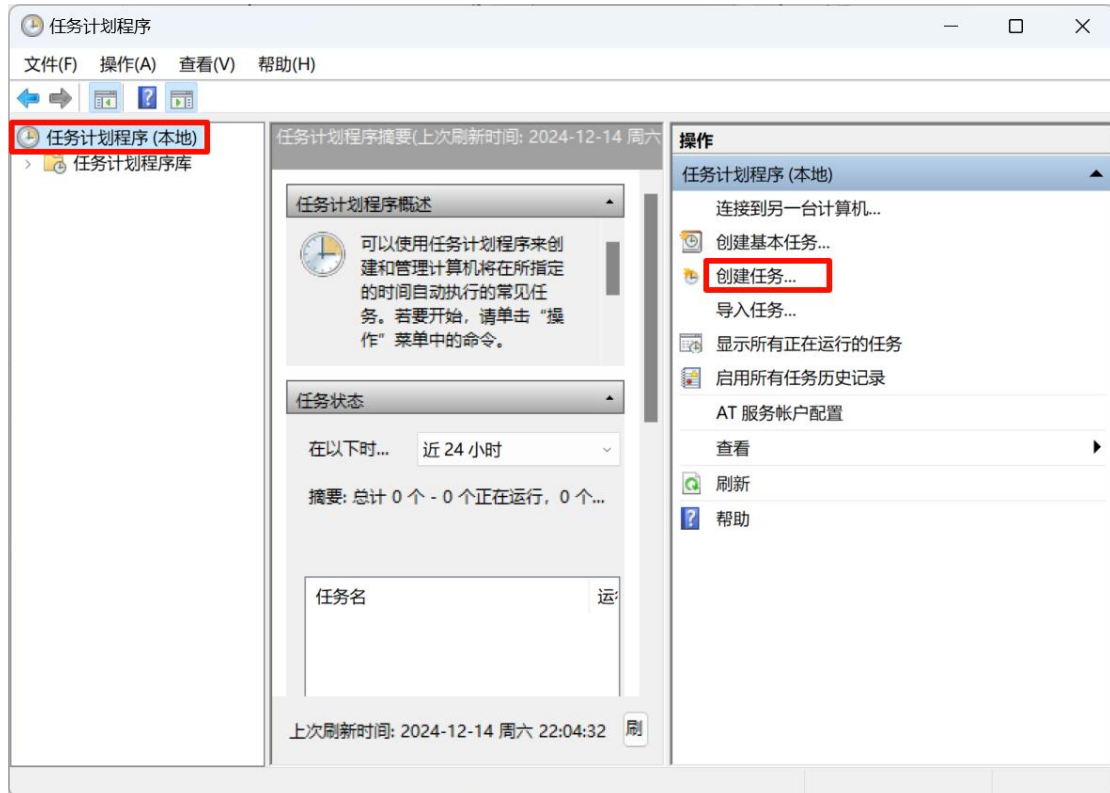


5. 设置自动同步

开始程序→搜索“任务计划程序”



任务计划程序（本地）→创建任务



输入任务名称和描述

创建任务

常规 触发器 操作 条件 设置

名称(M): Sync_Rime

位置: \

创建者: SZZHOUXING2\周星

描述(D): Rime自动备份

安全选项

运行任务时, 请使用下列用户帐户:

SZZHOUXING2\周星 更改用户或组(U)...

☒ 只在用户登录时运行(R)

☐ 不管用户是否登录都要运行(W)

☐ 不存储密码(P)。该任务将只有访问本地计算机资源的权限。

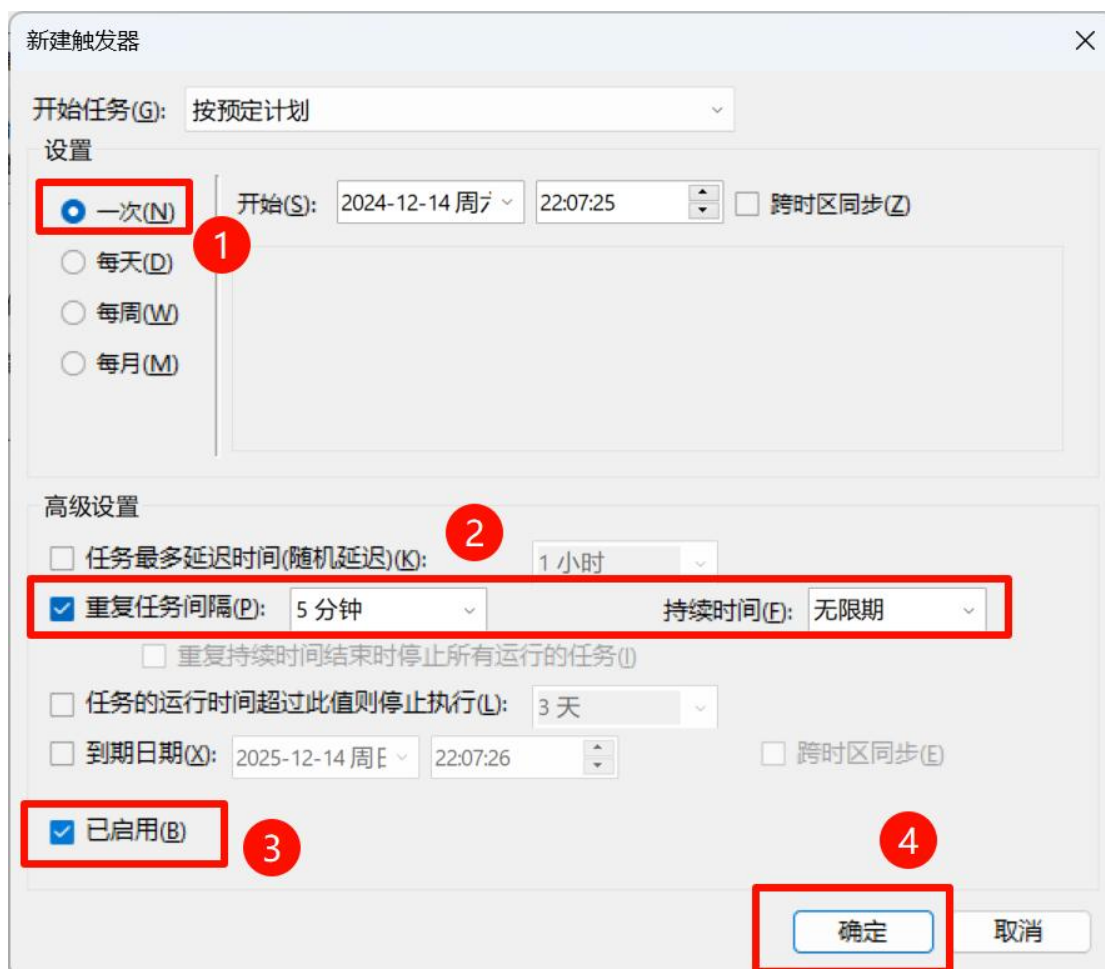
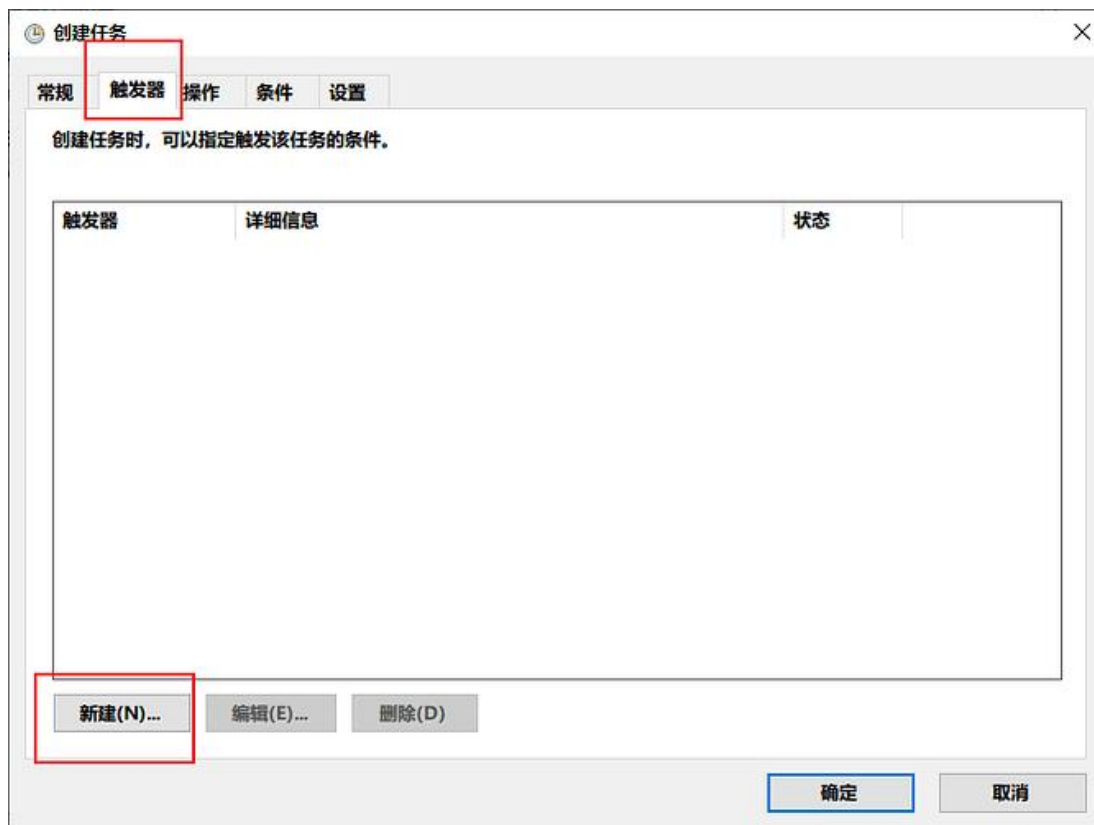
☐ 使用最高权限运行(I)

☐ 隐藏(E) 配置(C): Windows Vista™、Windows Server™ 2008

确定 取消

点击触发器→新建

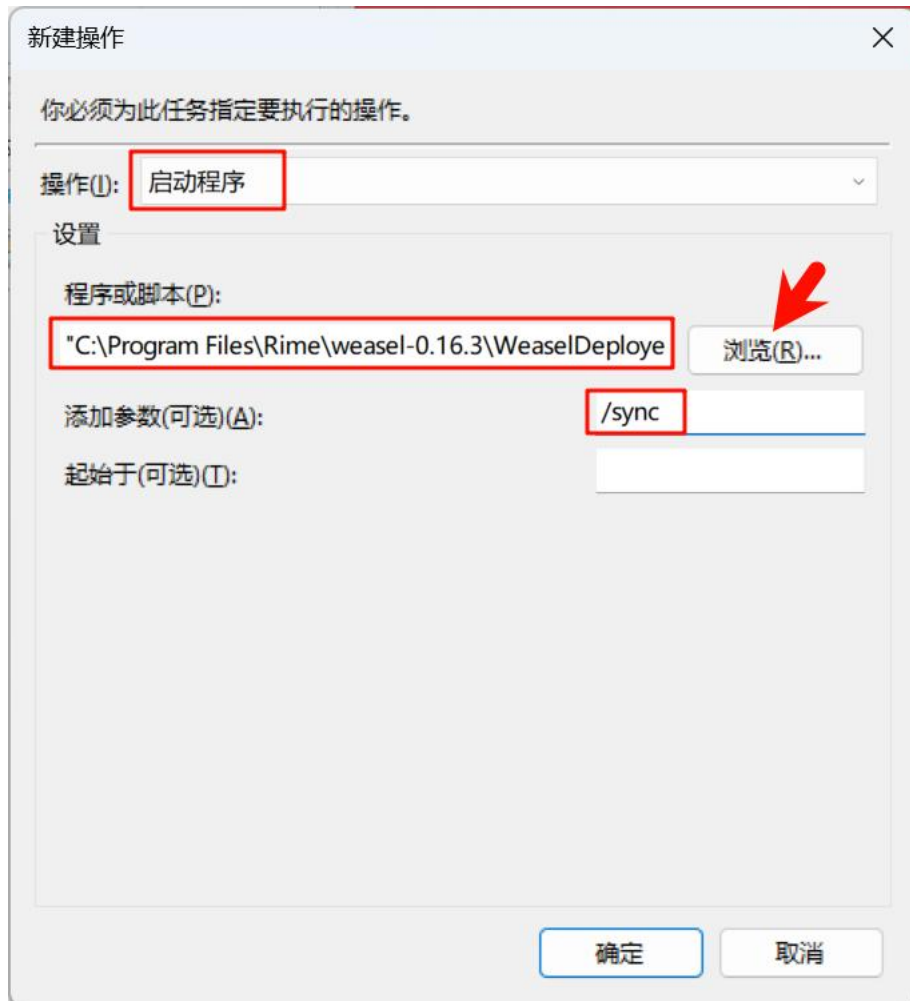
创建触发器，设置为每五分钟一次，无限期。



点击操作 - 新建

在操作中选择启动程序，程序或脚本选择 Rime 安装目录下的 WeaselDeployer.exe，参数添加 /sync

定时备份后，请检查 luna_pinyin.userdb.txt 的更新时间以确认是否更新。



6. 皮肤配色

现在开始配置很多朋友在意的「皮肤」，也就是 RIME 输入法的样式、布局与配色。

需要说明一点，RIME 之所以不称为「皮肤」，而是称为「样式、布局与配色」，在于：RIME 并不能指定背景图片。比如成品输入法皮肤市场各种各样动漫皮肤、卡通皮肤、风景皮肤，这并不支持。

RIME 只能配置输入法候选项的样式、布局和配色。

- **样式**涉及的方面有：候选框直排/横排、字体字号、单行模式/双行模式等等；
- **布局**涉及的方面有：候选框直角/圆角、候选框边距、候选项间距、阴影等等；
- **配色**涉及的方面有：候选框背景色、边框色，候选项色，待选项色，候选项高亮色，编码色等等。

在网上看到一个有趣的主题，如何添加到 Rime 中使用呢？以 Windows 平台为例，假如我们要加入[这里](#)提供的「丹青」主题，在系统 custom 设置文件（即 [weasel.custom.yaml](#)）中加入以下配置即可：

```
"preset_color_schemes/tantsing": # 在配色方案列表裏加入標識為
tantsing 的新方案
  author: Mijiag
  back_color: 0xE3E3E3
  border_color: 0x000000
  candidate_text_color: 0x000000
  comment_text_color: 0x474747
  hilited_back_color: 0x1A0791
  hilited_candidate_back_color: 0x6F0B73
  hilited_candidate_text_color: 0xE3E3E3
  hilited_text_color: 0xE3E3E3
  name: "丹青 / Tantsing"
  text_color: 0x000000
```

Rime 中，输入法的显示效果称为「颜色主题」(color scheme)，它已经默认带了一些主题，默认的主题在[这里](#)可以找到。要使用默认的主题，例如使用 lost_temple，只要在平台配置文件（例如 mac，就是 squirrelle.custom.yaml）中添加下面的配置：

```
patch:
  "style/color_scheme": lost_temple
```

7. 候选词的个数

修改 wubi86.custom.yaml 文件

```

patch:
# 设置应用默认中英文模式

switches/@0/reset: 0                #默认输入是中文还是英文，1为英文，0为中文
style/font_face: "微软雅黑"         #候选框字体
style/font_point: 25                #候选框字号
style/horizontal: false              #候选框是否横向显示
menu/page_size: 10                  #候选词个数
switcher/hotkeys: F2                #方案菜单快捷键

ascii_composer/good_old_caps_lock: true
ascii_composer/switch_key:
  Caps_Lock: commit_code             #Caps_Lock键切换大小写
  Shift_L: commit_code               #值为commit_code代表切换中英文，值为noop时代表不起作用
  Shift_R: commit_code               #值为commit_code代表切换中英文，值为noop时代表不起作用
  Control_L: commit_code             #值为commit_code代表切换中英文，值为noop时代表不起作用
  Control_R: commit_code             #值为commit_code代表切换中英文，值为noop时代表不起作用

key_binder/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

app_options: # 设置为在以下列表软件中，RIME 默认为英文模式
  cmd.exe: {ascii_mode: true}

```

8. 候选框是横还是竖

修改 wubi86.custom.yaml 文件

```

patch:
# 设置应用默认中英文模式

switches/@0/reset: 0                #默认输入是中文还是英文，1为英文，0为中文
style/font_face: "微软雅黑"         #候选框字体
style/font_point: 25                #候选框字号
style/horizontal: false              #候选框是否横向显示
menu/page_size: 10                  #候选词个数
switcher/hotkeys: F2                #方案菜单快捷键

ascii_composer/good_old_caps_lock: true
ascii_composer/switch_key:
  Caps_Lock: commit_code             #Caps_Lock键切换大小写
  Shift_L: commit_code               #值为commit_code代表切换中英文，值为noop时代表不起作用
  Shift_R: commit_code               #值为commit_code代表切换中英文，值为noop时代表不起作用
  Control_L: commit_code             #值为commit_code代表切换中英文，值为noop时代表不起作用
  Control_R: commit_code             #值为commit_code代表切换中英文，值为noop时代表不起作用

key_binder/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

app_options: # 设置为在以下列表软件中，RIME 默认为英文模式
  cmd.exe: {ascii_mode: true}

```

9. 候选词调频

10. 删除候选词

用 ↑ ↓ 选中候选框中想要删除的词，然后按快捷键 **Shift+Delete** 就可以删除候选词了。
系统词和自造词均可以用这种办法删除。

11. 模糊音配置

修改 pinyin.schema.yaml 文件

```
speller:
  alphabet: zyxwvutsrqponmlkjihgfedcba
  delimiter: " '"
algebra:
  - erase/^hm$/
  - erase/^m$/
  - erase/^n$/
  - erase/^ng$/
  - abbrev/^( [a-z] ).+$/ $1/
  - abbrev/^( [zcs]h ).+$/ $1/
  - derive/^( [nl] )ue$/ $1ve/
  - derive/^( [jqxy] )u$/ $1v/
  - derive/un$/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$2/
  # 下面两行实现 n、l 模糊音
  - derive/^n/l/
  - derive/^l/n/
  # 下面两行实现 in、ing 的模糊音
  - derive/([i])n$/ $1ng/
  - derive/([i])ng$/ $1n/
```

12. RIME 输入法的同步逻辑

13. 语法

词典的格式为: 词汇<Tab>编码<Tab>词频, 各个项目之间**必须用 Tab (也就是制表符)**分割, 刚开始我不知道, 结果使用了空格, 自己添加的词汇总是不成功, 需要注意。如果使用的是 **Sublime Text** 来编辑这个 **YAML** 文件一定要注意不要开启 `translate_tabs_to_spaces` 选项。词频部分可以不要。

一切添加妥当之后, 重新部署。然后试着输入, `up`, 候选列表会出现 `U 盘`, 输入 `yizhanghong`, 出现 `一丈红`, 说明词汇已经添加成功了!

Rime 没有 GUI 设置界面, 必需通过配置文件进行设置, 这点和 Vim/Neovim 类似, 它的配置文件使用 **YAML** 格式。每次更改配置文件以后, **都必需要重新部署**, 新的设置才能生效。

每次重新部署之后，最终生成的方案定义文件（原文件与自定义配置合并之后）`.schema.yaml` 会放在用户配置目录的 `build` 文件夹下面。我们可以打开里面的方案定义文件一探究竟。

请注意：`Patch:` 语句仅用于 `custom.yaml` 配置文档，整个配置文档有且只能有一个。请合理使用代码缩进进行对齐。

对于 `schema.yaml` 的修改是直接修改，可能会被更新覆盖；对于 `custom.yaml` 的修改是补丁修改，不会被更新覆盖，但该补丁文件需以 `Patch:` 开头，并且只能有一句。

Rime 的各种配置，均是由 `.yaml` 文件所定义。`yaml` 是一种标记语言。`.yaml` 文件实际上是文本文档。可使用记事本、或 `Emeditor` 等进行编辑。

对 Rime 进行自定义，是通过对 `.custom.yaml` 文件修改达成。不同的 `.custom.yaml` 文件，控制不同的功能实现。

`.custom.yaml` 实际上是相当于对 `.yaml` 文件打补丁，在重新部署后，会将 `.custom.yaml` 中的内容写入 `.yaml` 文件中，完成自定。

- 例一：**`weasel.yaml` 是常规设置**，主要控制托盘图标、候选词横竖排列、界面配色等等功能。那么，我们需要定制界面配色，只需在 `weasel.custom.yaml` 中修改，重新部署后就可实现。
- 例二：**`default.yaml` 是默认设置**，主要控制快捷键、按键上屏等等。同样，作修改就编辑 `default.custom.yaml` 文件即可。
- 例三：以上是全局设置，亦即不论使用何种输入方案，均起作用。`double_pinyin_flypy.custom.yaml` 这种则是输入法方案设置。主要实现特殊标点符号、词库等功能。是**针对特定输入方案**的配置。

可见，我们绝大部分的自定，都只需修改对应的 `.custom.yaml` 文件即可。

所有自定修改，都必须**重新部署**。在开始菜单可以找到**【小狼毫】重新部署**。

14. 为什么设置候选词竖排显示无效？

很多教程中说到，在 `weasel.custom.yaml` 里面使用如下的设置，使候选词竖排显示：

```
patch: "style/horizontal": false
```

但是部署以后，候选词还是横排显示，这是为何？可能是因为某些 `color_scheme` 本身把候选词设为了横排显示，主题设置的优先级更高，所以仍然横排显示，例

如 Rime 鼠须管有多款主题都在主题设置中把候选词设置为横排, 可以查看主题的配置来确定。

15. 问题解决

无法输入中文, 只能输入英文

解决方案是找到小狼毫安装目录, 找到 `WeaselServer.exe` 并固定到开始菜单, 遇到输入问题时, 运行该程序以重启算法服务。

16. 如何打出不认识的字

先按 F4 切换到[五笔画]输入法



然后按笔画顺序输入笔画即可, HSPNZ 分别代表横竖撇捺折。

例如“周”字可以用 PZHSHS 来打出

