

# xfold · ten-fold

## 1. 软件背景

- 日常使用电脑时，经常碰到需要一连串的按键和鼠标操作的使用场景。比如网页搜索，首先需要打开浏览器，打开百度搜索页，然后输入关键词后进行搜索。本软件通过定义使用命令行以及快捷键，快速完成上述繁琐过程，实现简化操作，节省时间（折叠时间）的目标。
  - 例如网页搜索，查询英语单词，查找本地文件，打开常用文件，保存摘抄笔记等等
- 软件名称由来：折叠时间 -> fold，提升效率 -> \*10 <==> xfold · ten-fold（十倍效率）
- 本软件主要作者自用，命令行及快捷键的定义，均使用文本编辑器修改扩展，可参考样例扩展配置，暂不提供配置界面。

## 2. 功能简介

xfold 是一个效率提升工具，支持将繁杂操作配置定义为快捷键或命令行，提供后续快速调用。

1. 扩展LButton组合键，双键，三键以及长按快捷键，定义快捷键有更多选择，有效缓解快捷键占用以及不容易记忆的问题。
2. 软件默认预置提供实用功能快捷键，比如，划词查单词，网页搜索，快速打开，笔记收藏等。
3. 内置基础功能操作有打开文件，目录，网址以及打开应用程序窗口，发送按键等，通过组合可轻松定义更多功能。
4. 参考示例进行简单修改，即可定义扩展更多的功能，并可为其指定快捷键或命令行。

### 2.1 特色功能说明

- 扩展LButton组合快捷键，更友好，告别热键冲突
  - 如：LButton+g，指按住鼠标左键先不松开，然后敲击键 g
  - 适用于鼠标划词或选择文本或文件时操作的场景，例如谷歌搜索选择的文本
  - ESC 或 \ 键可作为LButton替换按键，LButton+e 等同 ESC+e 等同 +e

注：当操作的窗口按Esc触发关闭或取消选择时，建议使用LButton+e或\+e

- 扩展双键，三键，长按快捷键，扩充出更多快捷键提供选择
  - Ctrl/Alt/Shift等组合快捷键在很多情况已被使用，容易遇到占用冲突
  - 扩展提供更多快捷键供选择，合理使用还能有助快捷键记忆
  - 扩展的键触发使用简单易用，如：双击F1，会比Ctrl+F1好用
- 支持特定应用内局部生效的快捷键

```
e e@chrome.exe => open(https://youdao.com/result?word={%x}&lang=en)
```

- 支持用户参数与变量，并提供友好输入
  - 系统环境变量，配置变量或用户输入等
  - %appdata% {%c} {%p} {%x}

- {%c} {c} : 剪贴板文本内容
- {%p} {p} : 用户输入的内容
- {%x} {x} : 发送Ctrl+C临时复制的内容
- 支持热字符串功能, 支持扩展定义
  - ]d: 输入框输入 "]d" -> 20240817 (输入日期)
  - ]w: 输入框输入 "]w" -> W202433 (输入第几周)
  - ]]: 输入框输入 "]" -> 20240817 周六 (输入日期 星期)
  - ..r: 输入框输入 "..r" -> xfold运行 输入框内文本
  - ..g: 输入框输入 "..g" -> xfold谷歌搜索 输入框内文本
  - ..v: 输入框输入 "..v" -> xfold获取变量值 输入框内文本

## 2.2 预置快捷键示例

- 常用搜索 Google - 选择文本时 LButton+g
- 常用搜索 Baidu - 选择文本时 LButton+b
- 常用搜索 Zhihu - 选择文本时 LButton+z
- 常用搜索 Douban - 选择文本时 LButton+d
- 英语查词 - 选择文本时 LButton+e
- 快速打开 - 选择文本时 LButton+r
- 文本编辑器打开 - 选择文本或文件时 LButton+a
- 复制窗口标题及选择内容 - LButton+t / LButton+t t / LButton+t长按
- 笔记收藏 - 选择文本时 LButton+s
- 在资源管理器中, 保存剪贴板内容为文件 LButton+Ctrl+v
- 在资源管理器中, 文本编辑器打开 Alt+e
- 在文件打开或保存对话框中, 跳转到当前资源管理器活动的目录 Ctrl+Q
- \* 快捷Everything查找文件 - 选择文本或文件时 LButton+f @依赖Everything
- \* Welink发消息 - 选择文本时 LButton+w
- \* W3查询人员信息 - 选择文本时 LButton+3 / LButton+w w

## 3. 下载及运行

1. 下载获取 xfold-x.x.x.zip 下载地址: <https://github.com/leexioua/xfold-dist/>
2. 解压软件包, 打开运行 xfold.exe

- 软件运行环境
  - + 必备: xfold.exe 或者 xfold.ahk (AutoHotkey.exe 脚本运行模式)
  - + 以下为推荐软件, 个人自用脚本涉及使用
  - + 推荐: sublime\_text.exe
  - + 推荐: Everything
  - + 推荐: QuickLook
  - + 推荐: 英语词典 eudic
  - + 推荐: FileLocator
  - + 可选: MouseInc
  - + 可选: CapsLock+
  - + 可选: QTTabBar
  - + 可选: welink
- 个人自用脚本参考
  - + 已打包保存 cmds\cmds-**for**-leexioua.zip
- 系统环境变量: 可选 (脚本运行模式需要设置)

```
+ Path=%Path%;D:\ProgramxFiles\AutoHotkey\AutoHotkey.exe
- 更多 >> 软件路径配置项: xfold.ini - [Settings]
```

## 4. 快捷键速查

- xfold-shortcuts.xlsx
- xfold-shortcuts.pdf

## 5. 快捷键功能扩展（高阶功能·普通用户建议跳过）

- 默认快捷键配置文件 keys-default.txt 如需更改为其它文件名称，可以在 xfold.ini - [Settings] - keymappings项修改
- 参考已有配置进行修改，映射包括快捷键及功能动作两部分 以 "=>" 分隔
  - 如 ^m => SafeActivate(%eudic%) 中
    - ^m 为快捷键， SafeActivate(%eudic%) 为功能动作
  - 如果同一快捷键被多次定义，只有最后面的定义生效，前面的定义不生效。即该快捷键不会触发多个功能动作，遵循覆盖更新原则。
    - 如果需要触发多个功能动作，可使用分号 (;) 分隔定义功能动作列表

### 5.1 快捷键定义

- 快捷键修饰符
  - # => Win 、 ^ => Ctrl 、 ! => Alt 、 + => Shift
  - 例： ^m => Ctrl+M #f => Win+F
  - Autohotkey 参考： <https://www.autohotkey.com/docs/v1/Hotkeys.htm#Symbols>
- 通用组合
  - 单键 single-press

```
^m => safeActivate(%eudic%)
```

- 双键 double-press

```
^q ^q => send(!{F4})
```

- 三键 treble-press

```
$F1 $F1 $F1 => open(notepad.exe)
; 此处按键F1前的$ 为避免多键快捷键的自我触发
```

- 长按 holding-press

重复四次及以上视作长按  
\$F1 \$F1 \$F1 \$F1 => alert("holding-pressed: F1")

- LButton组合键

```
LButton+a => edit  
LButton+g => open(https://www.google.com/search?q={%x})  
LButton+e e => open(https://youglish.com/pronounce/{%x}/english?)  
LButton+t t t => copyWinTitle  
LButton+t t t t => title
```

- 全局与特定应用生效

- 快捷键后跟 @应用进程名称 例: @explorer.exe

```
!e@explorer.exe => edit  
^e@sublime_text.exe => send(^p)
```

- 开关快捷键, 切换开关打开时生效

- 场景问题: e@chrome.exe => open(https://youdao.com/result?word={%x}&lang=en) 将会影响输入框输入字符e
  - 定义开关快捷键, 快捷键定义以“SwitchOn+”开头
    - 例: SwitchOn+e@chrome.exe => open(https://youdao.com/result?word={%x}&lang=en)
  - 使用时需要手动切换打开开关, 然后操作快捷键才能有效
    - 可使用默认快捷键: Ctrl+Shift+\
      - 切换开关快捷键修改 参考: \$F10 \$F10 => toggleSwitchOn
      - 应用启动时此开关为默认关闭, 需要手动切换打开

```
SwitchOn+e => eudicQuery  
SwitchOn+b => open(https://www.baidu.com/s?wd={%x})  
SwitchOn+d => open(https://www.douban.com/search?q={%x})  
SwitchOn+g => open(https://www.google.com/search?q={%x})
```

## 5.2 功能动作定义

常用功能动作直接调用内置的函数即可实现, 复杂功能可以自行编写脚本代码扩展。

- 常用功能动作
  - 打开 文件、文件夹、网页等 - open
    - open(cmd)

- cmd: 文件路径或网址, 或注册表路径
  - 示例: open(d:\file.txt)
  - 示例: open(C:\Windows)
  - 示例: open(http://www.baidu.com/)
  - 示例: open(http://www.google.com/search?q={%p})
  - 示例: open(HKEY\_CURRENT\_USER\Software)
  - 别名: run 即 run(cmd) 等同 open(cmd)
  - 注意: 如果打开失败, 会尝试调用funcCmd执行内置命令行高级功能。在自定义动作中可看到其应用
- 发送按键或文本 - send
  - send(keys, delay := 0)
  - keys: 按键或文本
  - delay: 延迟多少毫秒发送, 可选参数, 默认立即发送
  - 示例: send(^c)
  - 示例: send(how are you)
- 文本编辑器打开 - edit
  - edit(filename)
  - filename: 文件完整路径
  - 示例: edit(d:\file.txt)
  - 示例: edit(d:\file.txt)
  - 建议配置自己常用的文本编辑器, 默认为系统自带的记事本打开
    - 配置参考: 在xfold.ini中[Settings]下配置 editor=D:\ProgramxFiles\SublimeText\sublime\_text.exe
- 激活打开应用程序 - safeActivate
  - safeActivate(program, activate:=True, miniWhenActive := False)
  - program: 程序完整路径
  - activate: 是否需要激活应用程序窗口, 默认激活 取值: True或1, False或0
  - miniWhenActive: 当已经激活时是否将其最小化 取值: True或1, False或0
- 更多 参考内置动作函数列表
- 功能动作参数
  - open(C:\Windows) 中 C:\Windows 为参数
  - 多个参数以逗号分隔, 所以逗号为保留字符, 不能出现在参数值中, 为当前已知问题。
  - 所有参数以字符串传递。为简化使用, 参数可以不以引号包围, 即 open(C:\Windows) 等同 open("C:\Windows")
- 参数支持变量
  - 环境变量
    - 例: %appdata% %temp%
  - AHK运行环境变量
    - 例: %A\_ScriptDir% %clipboard%
  - 配置文件变量
    - xfold.ini中[Settings]定义的配置项 **优先**
    - xfold.ini中[Settings-default]定义的配置项
    - 例: %Everything% %eudic%
  - 特殊变量
    - {%c} 剪贴板内容
      - 等同 %clipboard%

- {c} 等同 {%c}

```
LButton+g => open(https://www.google.com/search?q={%c})
```

- {%p} 用户输入
  - 弹出对话框让用户输入
  - {p} {%q} {q} 等同 {%p}

```
LButton+g => open(https://www.google.com/search?q={%p})
```

- {%x} 发送Ctrl+C临时复制内容
  - 执行时发送Ctrl+C复制获取的文本内容
  - {x} 等同 {%x}

```
LButton+g => open(https://www.google.com/search?q={%x})
```

◦ 注意:

- 变量值中包含变量串也将被处理，代码只做两层引用替换
- 例：xfold.ini中[Settings]定义 MyLinks=D:\Links, home=%MyLinks%, 则 %home% 将处理替换为 D:\Links

- 多个功能动作
  - ;分隔多个动作

```
LButton+c c c c => send(^c);info(has been copied.)
```

- 内置动作函数列表

```
^(\w+)\(.*\)
使用参考: keys-default.txt, keys-leexioua.txt
```

函数	说明
alert(str, args*)	弹框提示
callback(key, cmd, params*)	注册回调函数 @see: recall recalls delaycb send sendif
clipValue(key)	设置剪贴板内容，取值value(key)
delay(delay, fn, params*)	延迟调用

函数	说明
delayif(delay, keyword, fn, params*)	延迟调用 满足特定条件情况时
delaycb(delay, cb, fn, params*)	延迟调用 调用完执行回调函数
getDerefStr(str, depth := 0)	用于变量值替换 例: %appdata% => getDerefStr("%appdata%")
getFileName(fullpath)	获取文件名称
getFilePath(fullpath)	获取文件路径
getSelectText()	获取选中文本
info(message, args*)	Tooltip提示
offDate(offset:=0)	获取日期字符串
open(cmd, param := "")	打开 文件、文件夹、网页
plainPaste()	不保留格式粘贴
preview(filename := "")	预览文件 参数为空时, 使用当前选择的文件名称
process()	获取当前活动窗口的进程名称
quoted(str)	加双引号
recall(key)	执行回调函数
recalls(keys*)	执行回调函数列表
redo()	重做最近的命令行命令
reload()	重新启动本程序
rstr(string)	提取行尾可执行字符串
run(cmd, param := "")	同 open
safeActivate(program, activate:=True, minWhenActive := False)	打开激活应用程序窗口
saveNotes(text := "")	保存文本到笔记, 参数为空时, 使用当前选择的文本
send(keys, delay := 0, cb := "")	发送按键或文本
sendif(keyword, keys, delay := 0, cb := "")	满足条件时发送, 检查匹配窗口标题, 进程名称是否包含关键词 例: sendif(exe:notepad.exe, 123) 例: sendif(title:标题, 123) or sendif(标题, 123)
show(keyword)	激活显示标题中包含指定文本的窗口, 有多个窗口匹配时显示最近打开的窗口。例: show(exe:notepad.exe) 例: show(title:标题) or show(标题)
sleep(delay)	中断等待, ms

函数	说明
stripLine(string, replace := "")	移除行首行尾相关字符，包括空格，回车换行，markdown项目编号及checkbox
title(removeTail := true)	获取当前活动窗口的标题
tooltip(str := "", ms := 2000)	Tooltip提示
urlDecode(url, enc = "UTF-8")	url编码
urlEncode(url, enc = "UTF-8")	url解码
value(key, val := "")	设置或读取用户变量key的值
vclip(key:="vclipboard")	临时保存的文本发送到剪贴板
vpaste(key:="vclipboard", delay := 0)	粘贴临时保存的文本
vselect(key := "vclipboard", section := "values")	保存所选择文本到用户变量key
vsend(key := "vclipboard", delay := 0, cb := "")	发送临时保存的文本
vsendif(keyword, keys := "v", delay := 0, cb := "")	当窗口匹配时，发送临时保存的文本
winmax()	最大化窗口
winmin()	最小化窗口
winrestore()	恢复窗口大小
winTopToggle()	置顶或取消置顶当前窗口
winTransparentDown(w := 0)	当前窗口增加透明
winTransparentUp(w := 0)	当前窗口减弱透明
xedit(filename := "")	文本编辑器打开
xfoldCmdline(cmd:= "")	打开命令行窗口
xsend(key, count:=1, lbtn:=false, title := "")	发送xfold快捷键,调用对应功能
xstr(key)	转码unicode, for ahk

- 自定义扩展动作
  - 在 xfold.ini - [Cmds] 自定义动作，默认调用open函数打开
  - 也可以通过此方式定义动作别名 例：echo=alert

[Cmds]  
 gi=https://www.google.com/search?q={%p}&tbm=isch



```
gmap=https://maps.google.com/maps?q={%p}&um=1
amap=https://ditu.amap.com/search?query={%p}
w3=http://w3.huawei.com/unisearch/contact.html?keyword={%p}
help=%A_ScriptDir%\readme@xfold.pdf
gtd=finditByEverything("ext:md dm:>=202001 path:D:\Workspace\D:\2Assort {%p}",0)
welink=send({raw}{%p},300);send(^!d)
cfg=edit(%A_ScriptDir%\xfold.ini)
set=cfg
msg=welink
echo=alert
```

```
; 使用参考
LButton+i => gi
或者
LButton+i => gi({%x})
```

### 5.3 热字符串扩展定义

- 配置定义：在 xfold.ini-[Hotstings] 中定义 例 ::@@=leexioua@gmail.com
  - 使用：在任意输入框输入“@@”后再空格或者Tab，@@被替换为“leexioua@gmail.com”
- 参考AutoHotKey文档
  - <https://wyagd001.github.io/zh-cn/docs/lib/Hotstring.htm>
  - <https://www.autohotkey.com/docs/v1/Hotstrings.htm>
- 参考示例

```
:o:@@=leexioua@gmail.com
:o:addr=Calle de xxxx, 28050 Madrid
```

### 5.4 配置文件 @xfold.ini

- 用户存储配置项信息，修改后需要重新启动脚本才生效
- 主要用于保存相关工具、脚本、帮助等文件路径信息
- [Settings-default] 下定义默认缺省的配置项的值
- [Settings] 下定义用户配置项的值
  - 获取参数时会 优先 获取 [Settings] 下对应配置项的值，如未找到，再查找获取 [Settings-default] 下对应配置项的值
- [Constants] 下定义常量，可被[Settings] [Settings-default] 引用

```
[Constants]
xfold=%A_ScriptDir%
ProgramxFiles=D:\ProgramxFiles

[Settings-default]
keymappings=%A_ScriptDir%\keys-default.txt;%A_ScriptDir%\keys-user.txt
```

```
editor=notepad.exe
previewer=notepad.exe

[Settings]
editor=%ProgramxFiles%\SublimeText\sublime_text.exe
```

## 5.5 配置实例

### 实例：豆瓣电影搜索

1. 获取分析URL
  - `https://search.douban.com/movie/subject_search?search_text=流浪地球2&cat=1002`
2. 参数拼接替换
  - 这里使用 `{%x}`，将当前窗口Ctrl+C复制的内容传入打开搜索
  - `https://search.douban.com/movie/subject_search?search_text={%x}&cat=1002`
  - 或使用 `{%p}`，调用时会让用户输入文本，用户输入的文本传入打开搜索
  - `https://search.douban.com/movie/subject_search?search_text={%p}&cat=1002`
3. 生成动作 此示例使用open
  - `open(https://search.douban.com/movie/subject_search?search_text={%x}&cat=1002)`
4. 定义快捷键映射
  - 快捷键映射配置文件中最后增加以下配置（`keys-default.txt` 或 在`xfold.ini`-[Settings]-Help\_keymappings 自定义的文件）
  - `LButton+d d => open(https://search.douban.com/movie/subject_search?search_text={%x}&cat=1002)`
5. 重启应用生效
  - 可使用内置默认快捷键(需应用已打开) `Ctrl+Alt+R`

#### • 注意

1. 如果出现多行定义了同一快捷键映射，以最后一行生效，即覆盖更新原则。  
快捷键定义： `keys-default.txt` 或 在`xfold.ini`-[Settings]-keymappings

2. 参数可以使用双引号或不加，以下两种方式写法均可  
`open(c:\windows\notepad.exe)`  
`open("c:\windows\notepad.exe")`

注：当前参数中英文分号（;）为保留关键字，在动作参数中使用可能引发问题，计划后续版本中将增加转义功能。

## 5.6 快捷键配置样例参考

- `keys-default.txt`

```
; 配置说明请参考 README.pdf
; 变量常量配置参考配置文件 xfold.ini
; 同一快捷键如果被定义多次，只使用最后一次定义，不会触发多个动作，即覆盖更新原则
; 修改配置后，需要重启xfold生效
```

^!r => reload ; 默认重启快捷键为: Ctrl+Alt+R, 可根据习惯修改为其它快捷键

```
;^1 => alert(ctrl 1)
;^1 ^1 => alert(ctrl 11)
;^1 ^1 ^1 => alert(ctrl 111)
;^1 ^1 ^1 ^1 => alert(ctrl 1111)
```

```
$` => delay(10,xfoldCmdline)
F1 F1 => delay(10,xfoldCmdline)
```

```
LButton+a => xedit({%x})
LButton+b => baidu({%x})
LButton+d => douban({%x})
LButton+e => youdao({%x})
LButton+g => g({%x})
LButton+i => gi({%x})
LButton+r => run({%x})
LButton+s => saveNotes
LButton+t => copyTitleTextLine
LButton+v => preview({%x})
LButton+y => you({%x})
LButton+z => zh({%x})
LButton+, => open(https://youglish.com/pronounce/{%x}/english?)
```

```
LButton+LAlt => send({Backspace})
LButton+Space => send({Enter})
LButton+Tab => send(!{Tab})
LButton++Tab => send(!+{Tab})
holdonLBtnWhen(LAlt)
holdonLBtnWhen(Space)
holdonLBtnWhen(Tab)
holdonLBtnWhen(+Tab)
```

```
LButton+1 1 => edit(%A_ScriptDir%\keys-user.txt);edit(%A_ScriptDir%\keys-
default.txt)
LButton+t t => copyTitleText
```

```
LButton+t t t => title
```

```
LButton+t t t t => title
LButton+p p p p => process
LButton+r r r r => run(rstr({%x}))
LButton+s s s s => xedit(%Link_notes%);send(^{end},500)
```

```
; @explorer
!c@explorer.exe => getCopyText
!e@explorer.exe => xedit
LButton+^v@explorer.exe => explorer_clipboardSave2Txt
```

```
; SwitchOn+ key definition
```

```
; ^!] => toggleSwitchOn
^!\ => toggleSwitchOn
SwitchOn+e@chrome.exe => open(https://youdao.com/result?word={%x}&lang=en)
```

- keys-user.txt

```
; 配置说明请参考 README.pdf
; 示例参考 keys-default.txt
```

```
LButton+2 => map({%x})
LButton+3 => w3({%x})
LButton+4 => redo
LButton+5 => term({%x})
LButton+6 => hwfy({%x})
LButton+7 => coa({%x})
```

```
LButton+w => welink({%x})
LButton+x => vo({%x})
```

```
LButton+c c => mdc
LButton+v v => vo({%x})
LButton+w w => hr({%x})
LButton+x x => x({%x})
```

```
LButton+^c => stripCopy
LButton+F12 => winTopToggle
```

```
#= => winTransparentUp
#- => winTransparentDown
^q ^q => send(!{F4})
^+v => plainPaste
```

```
explorer_dialogueJump(~^d)
```

```
; bak config
;LButton+w => send(^c);send(^!d,100);send(^v,300)
;LButton+f =>
clipboard({%x});open(shell:MyComputerFolder);send({F3},500);send(^v,1000) ;资源管理器默认搜索
;LButton+r r => term({%x})
;explorer_dialogueJump(~^q) ;文件对话框跳转到当前资源管理器活动的目录 Ctrl+Q 可根据需要修改 如Ctrl+D
```

- keys-leexioua.txt

```
; 个人使用快捷键配置，依赖相关软件工具及脚本 xfold.ini @leexioua
```

```

F1 => delay(10,xfoldCmdline)
F7 => redoHotkey

$` => send(#!q) ;capslock+
^$` => send(#!q)

^m => safeActivate(%eudic%)

#!n => safeActivate(%Evernote%)
~^!k => safeActivate(%keepass%,0)
#^!+f => safeActivate(%OUTLOOK%) ;mail-cmd.ahk中使用

$` $` => send({esc})
$` $` => xfoldCmdline

^e ^e => eudicQuery({%x})

^/ ^/ => setLBtnOn(3000)
^\\ ^\\ => toggleLBtnOn()
LButton+^\\ ^\\ => toggleLBtnOn()


F1 F1 => vselect;delay(10,xfoldCmdline);delaycb(500,vpaste,sendif,xfold-cmd)
F2 F2 => send(#!c) ;Yzime srf_create
F3 F3 => alist
F4 F4 => redo
F5 F5 => qlk;delay(200,max)
F6 F6 => qindex
F8 F8 => mail
F9 F9 => run({%x})
F12 F12 => setLBtnOn(3000)

;$` $` $` $` => xfoldCmdline
$` $` $` $` => send({Esc})

F1 F1 F1 F1 => send(#{F1})
F3 F3 F3 F3 => send(#{F3})
F10 F10 F10 F10 => getAccInfo

LButton+^c => stripCopy

LButton+c => collectNotes
LButton+e => eudicQuery({%x})
LButton+f => findit({%x}) ;everything搜索
LButton+h => song({%x})
LButton+j => jd({%x})
LButton+m => mail({%x})
LButton+n => noteit
LButton+q => q({%x})
LButton+v => previewx({%x})
LButton+y => you({%x})
LButton+$` => send({Esc})

LButton+p => send({Home})

```

```

LButton+; => send({End})
LButton+. => open(https://youglish.com/pronounce/{%x}/spanish?)

LButton+- => send({PgUp})
LButton+= => send({PgDn})
holdonLBtnWhen(-)
holdonLBtnWhen(=)

LButton+Left => send({WheelLeft 2})
LButton+Right => send({WheelRight 2})
LButton+Up => send({WheelUp 2})
LButton+Down => send({WheelDown 2})
LButton+Left Left => send({WheelLeft 4})
LButton+Right Right => send({WheelRight 4})
LButton+Up Up => send({WheelUp 4})
LButton+Down Down => send({WheelDown 4})
LButton+Left Left Left => send({WheelLeft 6})
LButton+Right Right Right => send({WheelRight 6})
LButton+Up Up Up => send({WheelUp 6})
LButton+Down Down Down => send({WheelDown 6})
LButton+Left Left Left Left => send({WheelLeft 8})
LButton+Right Right Right Right => send({WheelRight 8})
LButton+Up Up Up Up => send({WheelUp 8})
LButton+Down Down Down Down => send({WheelDown 8})
holdonLBtnWhen(Left)
holdonLBtnWhen(Right)
holdonLBtnWhen(Up)
holdonLBtnWhen(Down)

LButton+F2 F2 => open(D:\ProgramxFiles\BeyondCompare4\BCompare.exe)
LButton+F4 F4 => ihome
LButton+F11 F11 => adcTaskAdd
LButton+F12 F12 => ahkhelp

LButton+3 3 => cxkeys
LButton+4 4 => preview(%A_ScriptDir%\xfold-cmd-leexioua.csv)
LButton+6 6 => open(https://dida365.com/webapp/)
LButton+b b => bili({%x})
LButton+r r => delay(100,xfoldCmdlineRun,{%x})
LButton+v v => open(https://forvo.com/search/{%x})

LButton+F10 F10 F10 F10 => getAccAllText

LButton+1 1 1 1 => send(#{F1})
LButton+3 3 3 3 => send(#{F3})
LButton+c c c c => edit(%WORKSPACE_HOME%\#notes-list.md);send(^{end},500)
LButton+d d d d => md({%x})
LButton+f f f f => finditByEverythingFullPath({%x})
LButton+n n n n => tosave
LButton+q q q q => qst({%x})
LButton+v v v v => send(^c);edit(%temp%\vholder-xfold.txt);send(^{end}{enter}^v,
500)

```

```
LButton+v v v v => send(^c);send(!{tab},200);send(^{end}{enter}^v, 500)
LButton+w w w w => wx({%x})
LButton+x x x x => x {%x}

+Enter@Everything.exe => evcmd
;^Enter@Everything.exe => evcmd

/@SumatraPDF.exe => open(%Help_SumatraPDF%)
6 6@SumatraPDF.exe => send(^6)
7 7@SumatraPDF.exe => send(^7)
8 8@SumatraPDF.exe => send(^8)
e e@SumatraPDF.exe => eudicQuery

^2@sublime_text.exe => send(^r)
^d@sublime_text.exe => send(^+k)
^e@sublime_text.exe => send(^p)

LButton+1@sublime_text.exe => send(!c);delay(100,run,clipboard2cmd,previewx)
LButton+c c@sublime_text.exe => mdCheckToggle

!Up@ADC.exe => adc_execute(Move Task Up)
!Down@ADC.exe => adc_execute(Move Task Down)
!Home@ADC.exe => adc_execute(Move Task Top)
!End@ADC.exe => adc_execute(Move Task Bottom)
!Left@ADC.exe => adc_execute(Move Task Top)
!Right@ADC.exe => adc_execute(Move Task Bottom)

!e@FileLocatorPro.exe => send(!+c);sleep(200);edit({%c})

!c@WizTree64.exe => send(^!c)

-@QuickLook.exe => send(^{WheelDown 2})
=@QuickLook.exe => send(^{WheelUp 2})

Tab@eudic.exe => send(!{Tab})
-@eudic.exe => send(^!+-)
=@eudic.exe => send(^!+=)
RAlt@eudic.exe => send(^!+{Space})
LButton+e@eudic.exe => send(^ca);send({Backspace}^v{enter},300)

^q@cmd.exe => send(!{F4})

e e@Koodo.exe => eudicQuery

Tab@Everything.exe => showQuickLook({Tab})

+Esc@explorer.exe => send(+)=)

; 切换查看视图 @explorer.exe
```

```

LButton+1@explorer.exe => send(^!1)
LButton+2@explorer.exe => send(^!2)
LButton+3@explorer.exe => send(^!3)
LButton+4@explorer.exe => send(^!4)
LButton+5@explorer.exe => send(^!5)
LButton+6@explorer.exe => send(^!6)
LButton+7@explorer.exe => send(^!7)
LButton+8@explorer.exe => send(^!8)
holdonLBtnWhen(1@explorer.exe)
holdonLBtnWhen(2@explorer.exe)
holdonLBtnWhen(3@explorer.exe)
holdonLBtnWhen(4@explorer.exe)
holdonLBtnWhen(5@explorer.exe)
holdonLBtnWhen(6@explorer.exe)
holdonLBtnWhen(7@explorer.exe)
holdonLBtnWhen(8@explorer.exe)
Tab@explorer.exe => showQuickLook({Tab})

; potplayer plugin
XButton1 => send(^!+{Space})
^XButton1 => send(^!+{PgDn})
^XButton2 => send(^!+{PgUp})
+XButton2 => send(^!+-)
XButton2 => send(^!+=)

; configback plugin
#!F5 => configbak_Backup
configbak_MonthlyBackup()

; SwitchOn+ keys
SwitchOn+e@chrome.exe => en
SwitchOn+e@SumatraPDF.exe => en
SwitchOn+v@explorer.exe => xview({%x})

; matepad
; SwitchOn++d +d => send(#d)
; SwitchOn+1 1 => xsend(f1)
; SwitchOn+2 2 => xsend(f2)
; SwitchOn+1 1 1 1 => xsend(f1,4)
; SwitchOn+3 3 3 3 => xsend(f3,4)

delay(50,fnBackslash_toggleOnOff)
;delay(2000,toggleSwitchOn)

; bak config / samples
; \@eudic.exe => send(^!+\)
; ^[ ^[ => setLBtnOn(3000)
; ^] ^] => toggleLBtnOn()
; LButton+^] ^] => toggleLBtnOn()
;!c !c@sublime_text.exe => mdCheckToggle ;疑似引起!c冲突
;#` => restartAltRun

```



```

;#q => restartQuickLook
;$!Tab => send(!{Tab})
;$^Tab => send(^{Tab})
;$` $` $` $` => showQuickLook
;$` $` => send(!r)
;$` $` => showQuickLook;send({Esc})
;$F1 $F1 $F1 => toAltRun_xfoldCmd
;$F12 $F12 => alertLastCmd
;$F2 $F2 $F2 $F2 => markdown_clipboardImageURI
;$F2 $F2 $F2 => toAltRun_ahkCmd
;$F4 $F4 $F4 $F4 => gtd
;$F4 $F4 => redo
;$F7 $F7 $F7 => toAltRun_evernoteTerms
;$F8 $F8 $F8 => toAltRun_everythingTerms
;$F9 $F9 $F9 => toAltRun_mailCmd
;^+c@excel.exe => excel_BrushStart
;^2@sublime_text.exe => send(^p);send({U+0040},200) ; U+0040 => @
;^m ^m => en({%p})
;^RButton => clipboard({%x});send(^!+x,200)
;Browser_Back => send({Esc})
;Browser_Home => send(#d)
;F1 F1 => xfoldCmdline
;F1 F1 => delay(10,xfoldCmdline,{%x});delay(500,sendif,xfold-cmd,{end})
;F3 F3 => x({%p})
;LButton+$F12 $F12 => open(https://wyagd001.github.io/zh-cn/docs/index.htm)
;LButton+$F5 $F5 => edit(%A_ScriptDir%\xfold.ahk)
;LButton+1 1 => edit(%A_ScriptDir%\keys-user.txt);edit(%A_ScriptDir%\keys-
default.txt)
;LButton+2 2 => edit(%A_ScriptDir%\keys-leexioua.txt)
;LButton+5 5 => edit(%A_ScriptDir%\xfold.ini)
;LButton+8@sublime_text.exe => sublimetext_convertToUTF8
;LButton+a a a a@sublime_text.exe => viewAhkFile({%x})
;LButton+e => clipboard({%x});send(^v{enter},300);safeActivate(%eudic%) ;配置方式实
现英语查词
;LButton+Left => send({PgUp})
;LButton+m m@sublime_text.exe => sublimetext_markdownSyntax
;LButton+q => evernoteSearch
;LButton+r r r r => run({%p})
;LButton+Right => send({PgDn})
;LButton+x x x x@sublime_text.exe => mdCheckToggle
;SwitchOn++,@PotPlayerMini64.exe => send(x)
;SwitchOn++,@PotPlayerMini64.exe => send(c)
;SwitchOn++/@PotPlayerMini64.exe => send(z)
;SwitchOn++n@PotPlayerMini64.exe => send(^!+{PgDn})
;SwitchOn++p@PotPlayerMini64.exe => send(^!+{PgUp})
;SwitchOn+f => alert(f1)
;SwitchOn+f f => alert(f2)
;SwitchOn+f f f => alert(f3)
;SwitchOn+f f f f => alert(f4)
;SwitchOn+f@PotPlayerMini64.exe => send({Enter})
;SwitchOn+j@PotPlayerMini64.exe => send({Left 2})
;SwitchOn+k@PotPlayerMini64.exe => send({Space})
;SwitchOn+l@PotPlayerMini64.exe => send({Right 2})
;SwitchOn+LButton+f => alert(lbtn+f1)

```

```
;SwitchOn+LButton+f f => alert(lbtn+f2)
;SwitchOn+LButton+f f f => alert(lbtn+f3)
;SwitchOn+LButton+f f f f=> alert(lbtn+f4)
;Tab Tab => send(!{Tab})
```

## 6. 命令行功能扩展（高阶功能·普通用户建议跳过）

- 软件预置提供了部分命令行，例如 `g cmds keys` 等
- 预置的命令行在 `xfold.ini` 配置文件中定义，参考 `[Cmds-default]` `[Cmds]` `[Cmds-alias]`
- 如果有重名定义，优先级依次 `[Cmds]` > `[Cmds-default]` > `[Cmds-alias]`

### 6.1 配置文件命名规范

- 默认使用 `xfold.ini` 扩展配置项即可
- 如果需要模块拆分方便维护，可以按照 `xfold-*.ini` 规范命名
- 多个配置文件时加载顺序按 文件名倒序加载，已有的配置项目不做覆盖更新，按照此约定 `xfold.ini` 中的配置项优先级最级。

### 6.2 命令行配置

- 命令行配置包含名称和命令行两部分
- 名称要求为字符和数字加连字符，例如 `noteit w3 gtd-inbox`
- 命令行参考 5.2 功能动作定义
  - 支持命令行的写法 `g({%p})` 与 `g {%p}` 等同
  - 支持包含多条命令 以 ; 分隔 例如: `b({%p});g({%p})`
- 示例:
  - `baidu=https://www.baidu.com/s?wd={%p}`
  - `bg=b({%p});g({%p})`
- `[Cmds-default]` 建议默认常用命令行在这里定义。
- `[Cmds]` 建议自定义扩展命令行在这里定义。
- `[Cmds-alias]` 命令行可以设置别名，建议在这里定义。

### 6.3 命令行使用

- **命令行窗口**
  - 默认使用 ``` 或 `F1 F1` 打开命令行输入框，输入命令行回车即可执行命令

```
; defined in keys-default.text
$` => delay(10,xfoldCmdline)
F1 F1 => delay(10,xfoldCmdline)
```

- 命令行中的参数输入，按以下输入格式都可以执行
- `g table`
- `table.g`
- `table>g`

- **快捷键** LButton+r 选择的文本作为命令行执行

```
; defined in keys-default.text
LButton+r => run({%x})
```

- **魔法字符串** ..r 任意输入框输入命令行，最后连续追加..r

- g table..r

```
; defined in xfold.ini
:*?xo:..r=zmd:send(+{home}^c{end});delay(100, xfoldCmdlineRun, {%cx})
```

## 7. 贡献

- 感谢Esther设计应用图标

## 更新日志

---

### 遗留优化 TODO

- 增加定义快捷键查找功能 @xfold-shortcuts.xlsx
- 与CapsLocks+冲突问题，如 CapsLock+F1
- 解析热键定义中的保留字符使用 ; , - 通过anltr词法解析
- 增加组合键扩展 LButton+a e / Ctrl+k, Ctrl+d - 思路 hotstring-zmd，并击
- 增加代码注释自动生成文档 类似jsdoc AHKDoc
- 自动化测试/用例
- 打包后支持自定义脚本扩展功能 LexAHKL.dll
- 迁移到Autohotkey2.0
- funcCmd与open区分使用，funcCmd 标准函数方式，open在funcCmd扩展支持命令行及路径
- 插件 cmd+hotkey+hotstrings
- 整理xfold-base.ahk 函数有些乱
- 项目工程重新构建

### @issue

- xfold.ini 需要保存文件编码 UTF-16 LE with BOM
- xfoldCmdlineRun中以.md类似结尾误解析为命令动作，例如 f xfold设计思想.md - 暂命令在最后添加空格规避
- 使用虚拟键盘输入 设置了快捷键的字符无法输入 - 原因：虚拟键盘输入时LButton为按击状态 LButton+f。 getLBtnState - 通过开关规避解决 toggleVLBtnEnabled
- ; , 为保留字符，命令行中参数包含时存在错误解析

### 20240916: 1.2.2

- 新增重做上次执键功能 redoHotkey()

- 新增打开应用或文件时显示已打开 ashow() - "assure-show"
- 新增执行批量命令 runs/runcmds
- bugfix d:\not-exist-file.txt
- 解决编译exe无法运行问题
- 修改默认日志级别: error
- 清理硬编码的hotstings - spanish-accents
- xfold-cmd cmd增加连字符的支持 gtd-in
- 修改增强使用run命令执行对应动作 for xmd.exe keys-\*.txt
- 整理配置文件

## 20240818: 1.2.1

- 增加支持多配置文件，默认: xfold.ini, 扩展配置命名: xfold-\*.ini 加载顺序按名称倒序
- 整理代码并修复部分问题

## 20240609: 1.1.12

- 解决命令行窗口只能呼出一个的问题 delay(10,xfoldCmdline)
- xfold-cmd命令行窗口, ctrl+enter vgo, shift+enter 使用选择文本填充参数

## 20240526: 1.1.11

- 增强send功能, 增加支持回调callback功能
  - 使用示例send({space}^v@,500,strd);callback(strd,send,offDate())
- 增加支持在任意输入框中, 快速执行命令行 ..r
  - .\*?xo:..r=zmd:send(#{home}^c{end});delay(100, xfoldCmdlineRun, {%c})
- 增加支持快速输入系统或配置变量的值 ..v
  - .\*?xo:..v=zmd:send(#{home}^c{end});delay(100, sendvcfg, {%c})
- 规避使用包含特殊字符作为参数的问题
  - 增加从文件临时保存提取字符串功能@xfold-value.ahk vget, vset, vappend, vsetarr
  - 增加当前选择文本临时保存 vselect(v)
  - 增加发送临时保存的字符串 vsend(v)
  - 增加将临时保存的字符串写入剪贴板, 供后续粘贴使用 vclip(v)
- 修正xfoldCmdline输入不存在的命令时的处理错误
- 重新打包xfold.exe

## 20240218: 1.1.10

- 增加虚拟键LButton按下及释放功能, 无鼠标时也可使用LButton+e等功能
  - 例: \$F12 \$F12 => setLBtnOn(2000) 临时按下虚拟键LButton, 2秒后释放 @keys-leexioua.txt
  - 例: ^/ ^/ => setLBtnOn(2000) 临时按下虚拟键LButton, 2秒后释放 @keys-leexioua.txt
  - 例: ^\ ^\ => toggleLBtnOn() 保持按下虚拟键LButton @keys-leexioua.txt
  - 例: LButton+^ \ ^\ => toggleLBtnOn() 释放虚拟键LButton @keys-leexioua.txt
- 增加窗口最大化、最小化、还原、窗口置顶的命令行功能
  - max,min,restore,top
- 延迟调用delayCall增加别名delay

- `preview(%A_ScriptDir%\xfold-cmd.csv);delay(300,showQuickLook);delay(500,winmax)`
- 命令行窗口执行命令时，按Shift或Ctrl键+Enter，自动复制选择文本内容作为用户输入的参数执行命令行
- 增加SpaceFn, BackslashFn插件功能，Space或\键代替LButton功能。生效切换开关Ctrl+Win+Space, Ctrl+Win+\
- 增加hotstring-cmd功能，可支持特定应用内生效
  - 例 `:*xo:zmdyw=zmd:send(%A_YWeek%)`
  - 例 `:*xo:zzf=zmd:f {%x}`
  - 例 `:*b0:hip@chrome={bs 3}122.96.12.2`
- 增加Everything-cmd功能，在Everything 输入框中使用组合键Ctrl/Shift+Enter可执行xfold cmdline
- 调整部分热键设置 @leexioua.txt
  - SwitchOn+ 增加PotPlayer支持Youtube常用热键
  - 其它调整
- 整理补充更多命令行方法 @xfold.ini
  - 窗口相关操作等

## 20231005: 1.1.9

- 增加临时LButton按下功能setLBtnOn，不使用鼠标左键也可完成其组合键功能，如 LButton+f
- 增加cmd参数支持，支持引用上一次临时获取的字符内容 {%} 或 {%last}
- 增强打开功能，支持提取结尾的路径执行打开，LButton+r r r r => run(rstr{%x})) @keys.user.txt
- 解决长时间不按键再次按键误判问题，增加2s间隔条件
- 其它优化

## 20230708: 1.1.8

- xfold-cmd 优化命令处理优先级，配置项自定义动作 > 内置动作 > 系统cmd
- 优化参数变量读取优先级 用户value() > 配置项 > ahk运行变量 > 环境变量
- 资源管理器保存剪贴板内容文件名支持配置，配置项名 QuickSaveFileName 默认值 readme@xfold.md
- 增加sendif(keyword, keys, delay := 0) 满足条件才发送，检查匹配窗口标题，进程名称是否包含关键词
  - 例： `sendif(exe:notepad.exe, 123)`
  - 例： `sendif(title:标题, 123) or sendif(标题, 123)`
- 修正send延时发送包含特殊字符时一直发送的问题
- 托盘菜单增加打开xfold-dist@github项目首页的菜单项
- 配置优先级应高于系统应用解决 edit D:\Links\ahk\xfold\xfold-base.ahk
- 调整优化默认快捷键定义 @keys-default.txt, @keys-user.txt
- 调整优化cmds定义 @xfold.ini

## 20230528: 1.1.7

- 增加 xsend(key, count:=1, lbtn:=false, title := "") 发送xfold快捷键,调用对应功能
- 优化Cmds定义 @xfold.ini。加载逻辑同Settings，优化加载读取 [Cmds]，未读取到再加载[Cmds-default]
- 优化Hotstrings定义 @xfold.ini。加载逻辑同Settings，优化加载读取 [Hotstrings]，未读取到再加载 [Hotstrings-default]
- 增加QuickLook窗口激活功能 例： `Tab@explorer => showQuickLook({Tab})`
- 增加show(title) 激活显示标题中包含指定文本的窗口，有多个窗口匹配时显示最近打开的窗口
- 增加xstr(str) 转义相关保留字符, ",", ";", "@", "(", ")"

- 增加xfold-cmd命令行功能，支持以下命令格式
  - cmd: g table
  - cmd pipe: table>g
  - function: g(table)
- 整理 xfold-cmd 命令行参考 @xfold-cmd.csv
- xfold-cmd打包命令行工具 xmd.exe @xfold-cmd.ahk
- 更新打包应用的icon
- 补充调整快捷键定义 @keys-leexioua.txt
- 补充调整Cmds定义 @xfold.ini
- Esc代替LButton快捷键，未知原因失效 - 已恢复

## 20230514: 1.1.6

- 增加在资源管理器中复制路径功能 !c@explorer.exe => getCopyText
- 移除其它可直接配置实现的插件
- 优化发送命令行功能 xfold(cmd)
- 修复可执行程序版本找不到图标文件错误

## 20230512: 1.1.5

- 增强xfold-cmd，检测xfold脚本或可执行程序运行，提供外部发送命令调用接口 例：xfold-cmd.exe en table
- 增加脚本编译可执行选项设置
- 增加托盘菜单，菜单增加重新加载配置及关于功能
- 快捷键定义解析处理优化，读取行后移除行尾注释(行最后包括空格分号的字符串)  
例：!c@sublime\_text.exe => xxx ; copy path@sublime\_text
- 移除seer插件

## 20230508: 1.1.4

- 增加后台运行打开功能 命令行以" -hidden" 例：open("xxx -hidden")
- 增加 delayCall 支持动作方法的延迟调用 例：delayCall(2000, "send", "!r")
- 增加开关型扩展快捷键，需要在手动打开开关，快捷键才生效，手动关闭后，快捷键失效。场景问题：  
e@chrome.exe => open(https://youdao.com/result?word={%x}&lang=en) 将会影响输入框输入字符e
  1. 定义开关型快捷键，快捷键前加"SwitchOn+" 例：SwitchOn+e@chrome.exe => open(https://youdao.com/result?word={%x}&lang=en)
  2. 使用时需要手动打开开关。可使用默认快捷键: \$F1 \$F1 或者 Ctrl+Shift+\\ (开关快捷键修改参考: \$F10 \$F10 => toggleSwitchOn)
- 增加热字串的配置功能
  1. 在 xfold.ini-[Hotstings] 中定义 例 ::@@=leexioua@gmail.com 输入"@@"后再空格，替换为 leexioua@gmail.com
  2. 参考AutoHotKey文档 <https://wyagd001.github.io/zh-cn/docs/lib/Hotstring.htm> (或 <https://www.autohotkey.com/docs/v1/Hotstrings.htm> )
- 修正优化stripLine问题，移除行首markdown项目符号
- 修正同一快捷键在多个特定应用中只会在第一应用中生效的问题 bindKey

## 20230423: 1.1.3

- send 增加延时发送 setTimer 例: send(%clipboard%, 1000)
- {%p} 参数记录用户最后一次输入, 作为输入框默认值
- funcCmd 支持嵌套使用 例: alert(title())
- 增强 funcCmd 支持函数命令行形式的CMD调用 例: en desk <=> en(desk)
- 增强 funcCmd 参数传递{%p}使用, 不需要用户交互输入 例: en({%p}): en desk
- 增加自定义命令功能:
  1. 在 xfold.ini-[Cmds] 中定义 例 gi=https://www.google.com/search?q={%p}&tbm=isch
  2. 快捷键配置 LButton+i => gi, 或者传递参数方式 LButton+i => gi({%x})
- Esc 替代 LButton 逻辑优化, 按键后不释放
- 开放xfoldCmd(cmd) ==> funcCmd(cmd) 给外部脚本调用, 例: xfoldCmd("send(!r)")
- 增加 invokeKeyCmd/getKeyCmd 方法, 支持直接执行快捷键动作
- 配置项对快捷键配置增加支持多文件叠加, 以分号 (;) 分隔。例: keymappings=%A\_ScriptDir%\keys-default.txt;%A\_ScriptDir%\keys-leexioua.txt
- 提升发送级别, 可发送本脚本监听快捷键 -> SendLevel 1
- 部分日志清理为 debug 日志
- 其它优化

## 20230413: 1.1.2

- 增加 value(key, val) 提供保存用户变量, 以便后期使用。例: LButton+w => value(hwid, {%x});open(notepad);send({Text}%hwid%, 2000)
- 修正 F8长按失效问题 用例: \$F8 \$F8 \$F8 \$F8 => open(autohotkey.exe D:\Links\filelocator\locator-cmd.ahk {%p})
- 修正 资源管理器中保存剪贴板内容不信赖标题显示完整路径设置 @explorer\_clipboardSave2Txt
- 增强 在资源管理器中 Alt+e 文本编辑选中文件 例: !e@explorer.exe => edit({%x})

## 20230409: 1.1.1

- 通过配置文件定义热键绑定, 方便后续 ahk2exe 打包可执行文件发布。
  - 默认热键映射配置文件 keys-default.txt 如需更改为其它文件名称, 可在 xfold.ini - [settings] - keymappings 中修改

## 20230408: 1.1.0

- 项目取名 xfold, 修改相关文件及目录命名, xfold初始版本为1.1.0
- 增加配置文件常量定义, 变量值中支持嵌套变量

```
[Constants]
ProgramxFiles=D:\ProgramxFiles
[Settings]
editor=%ProgramxFiles%\SublimeText\sublime_text.exe
```

- 增加重做最近一次动作 redo, 类似excel F4功能。热键: F4 F4 => redo, F12 F12 => alertLastCmd
- ahk脚本实现收集保存内容功能 saveNotes。LButton+s => saveNotes

- open 动作支持快速打开注册表路径 依赖 regjump.bat 例:  
open(HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Applets\Regedit)
- 个人自用脚本打包供参考 cmds\cmds-by-leexioua.zip
- 修复Esc存在多次发送问题 @xfold-init-keybinding.ahk
- 调整应用框架代码, 重命名, 目录位置
- 修改命名 holdLBtnAfterAction => noReleasingLBtnAfterAction

## 20230330: 1.0.11

- 整理简化不必要的动作方法 @leexioua-actions.ahk
- 增强funcCmd支持ahk系统变量 例 %A\_ScriptDir%
- 增加 LButton+c => Ctrl+c, LButton+x => Ctrl+x, LButton+v => Ctrl+v, LButton+z => Ctrl+z
- 支持默认及用户键映射定义扩展 例 @leexioua-keys-default.ahk @leexioua-keys-sean.ahk
- 整理默认通用键映射配置 减少依赖 @leexioua-keys-default.ahk
- 修改自用键映射配置文件命名 @leexioua-keys-sean.ahk
- 优化 open 动作, 打开网页时参数拼接, 参数前不补空格 open("https://www.google.com/search?q=", "{%x}")
- 优化 open 动作, 捕获运行异常并提示用户
- 修改配置文件编码为UTF-16 @leexioua-ahk.ini
- 整理脚本入口, 分离基础与个性化的功能, 以便后续使用用户自主选择或扩展增强功能 @leexioua-ahk.ahk
- 优化 stripLine, 增加 # ; 等引导符剔除
- 优化增加默认参数 [Settings-default] @leexioua-settings.ahk

## 20230324: 1.0.10

- open(cmd)/funcCmd(cmd) 增加支持临时复制选择文本的参数 {%x}
- 增加 title(removeTail:=True), 获取当前窗口的标题到剪贴板
- 增加 process, 获取当前窗口的进程名到剪贴板
- 增加 LButton+r长按 => run({%p}), 支持运行内置的动作
- 增加 plugin-markdown.ahk 支持将剪贴板图片/图片路径, 复制成markdown DataURI形式 ![pic]()
- 修正funcCmd(cmd) {%p}参数不生效问题
- open 增加别名 run, open(notepad) <==> run(notepad)
- 修改命名 holdingLBtnAfterAction => holdLBtnAfterAction

## 20230318: 1.0.9

- 增加支持长按热键, 热键定义键数4次以上认为是长按绑定定义 @leexioua-keybinding.ahk 用例:  
LButton+n n n n => open(notepad)
- open(cmd)/funcCmd(cmd)增加支持用户输入参数{%p}与剪贴板文本{%c}, 例:  
http://www.google.com/search?q={%p}
- open(cmd) 增强扩展其功能, 调用Run失败会再尝试调用funcCmd方法
- 增强 funcCmd (cmd) 热键可支持执行多个函数 用例: r r@notepad.exe =>  
alert(123);info(1334);alert(333)



- 增加 holdingLBtnAfterAction(key, enable), 用于控制是否在执行动作后释放{LButton up} 用例:  
holdingLBtnAfterAction("1@explorer.exe", true)
- 增加插件 plugin-configbak 用于备份个人使用软件的相关配置文件
- 增加 ctrl+e e => eudic
- 引入日志模块, 方便问题定位
- 优化修改击键弹起触发热键, 即按键释放时触发
- 优化当只有单个按键热键, 不需要等待下一次按键判断, 单键热键动作无延时 用例:  
^d@sublime\_text.exe => send(^+k)
- 优化扩展相关动作参数, 无参数时使用复制选择内容/剪贴板内容 例: eudicQuery(word:= "") @ leexioua-actions.ahk
- 清理可简单使用funcCmd()实现的方法 @leexioua-actions.ahk 例: keymapping() =>  
edit(%Help\_keymappings%)
- 增加依赖到脚本目录下的Lib Acc.ahk, Anchor.ahk
- 修正funcCmd (cmd)参数解析问题
- LButton+r => openOrRun 优化支持环境变量, 用户输入参数

## 20230314: 1.0.8

- 配置参数名称优化, 统一使用getCfgValue()
- 验证已支持组合键扩展 (Ctrl+c c) 用例: \$^o \$^o => SafeActivate(%eudic%)
- 相关插件注册热键改用bindHotKey()
- 修正bindHotkey不支持LButton+w w问题
- 键映射调整, 参考 leexioua-shortcuts-ahk.xlsx
- 修正WinActive title参数错误 用例: r r@notepad.exe => alert(123)

## 20230307: 1.0.7

- 增加热键 ` => ALTRun `` => clQBar@CapsLock+
- 支持同一热键在不同应用中执行不同的动作 LButton+f f@sublime\_text.exe => findIt, LButton+f f@notepad.exe => alert(123)
- 动作支持变量, 系统环境变量 或配置项的值 例: SafeActivate(%Path\_eudic%), eudicQuery(%clipboard%)
- 支持Esc+单键, 双键与三键, 等同LButton键组合, Esc+f => LButton+f
- 重构 keybinding, 将plugin-keymapping 合并
- 修正绑定单键热键时, 字符输入失效问题 1 1 => open(notepad)
- 修正插件与keybinding插件热键注册冲突问题

## 20230303: 1.0.6

- 重写配置加载模块, 启动时动态加载配置, 简化了新增参数需要使用的代码行
- 部分键绑定调整 leexioua-keys-keymappings.ahk
- 其它优化增强

## 20230221: 1.0.5-beta

- keymapping配置支持使用含参数的方法调用 例: open(notepad.exe),  
edit(D:\ProgramxFiles\MouseInc\MouseInc.json)

- 新增公共方法供keymapping配置使用，打开程序/文档/网址 open(cmd), 文本打开编辑文件 edit(filename)
- 增加keepass插件，功能检查并确保keepass已打开运行
- 修复SafeActivate() 等待窗口打开后发送按键功能
- CapsLock+单键 绑定带参数的方法错误, 例 LButton+x => open(xxx)
- 取消 CapsLock+单键 => LButton+单键功能，预留CapsLock 作其它用途
- 改写标签方式的用的热键功能实现，修改为插件方式实现
- 提高发送按键的层级优先级，确保外部ahk应用热键可接收到
- 修正由于插件加载顺序引起keymapping不能注册成功
- 按键名称转义发送效率优化 escKeys4Send(A\_ThisHotkey)
- 增加markdown快速标记复选框完成状态 暂使用 LButton+x x => @sublime text

## 20230214: 1.0.4

- 新增插件简化热键配置 plugin-keymapping.ahk , 如: LButton+e e
- 修改资源管理器文件对话框中热键 Ctrl+Q (快速跳转Quick)
- 其它优化，复制文本预处理使用统一方法

## 20230210: 1.0.3-beta

- 新增配置项集中管理 leexioua-ahk.ini
- 新增ahk-cmd.ahk 动态ahk脚本发送到AltRun
- 新增资源管理器文件对话框中 Ctrl+Alt+D 快速跳转到当前活动的目录

## 20230208: 1.0.2

- 新增知乎搜索 LButton+z
- 新增searchAhkSource - F8双键
- 新增LButton+Tab -> Alt+Tab 切换窗口 @leexioua-lbl-global.ahk
- 新增打开查看leexioua-shortcuts-ahk.xlsx (快捷键帮助), LButton+F1
- 新增保存资源管理器剪贴板文本到文件 readme@leexioua.md @leexioua-lbl-explorer.ahk - 需要资源管理器选项设置 在标题栏中显示完整路径
- 新增格式刷热键 Ctrl+Shift+c @leexioua-lbl-excel.ahk
- 新增everything-cmd, evernote-cmd, mail-cmd 搜索关键词发送到AltRun, ahk-func中方法发送到AltRun
- leexioua-actions 修改相关方法名称

## 20230203: 1.0.1

- 修正viewAhkFile 路径拼接错误
- 修正plugin-keysfunc调用异常，键失效问题
- 修正plugin-keysfunc countKeys SetTimeout 偶尔绑定失败后，键失效问题
- 新增 copyTitleText, copyTitleTextLine for LButton+t t

## 20230127: 1.0

- 代码重新整理，模块化，规范代码，按功能和应用结构分享代码；插件加载方式方便根据需要修改加载

- 增加多键热键支持，扩展更多可定义的热键 LButton+多键（单键，双键，三键）
- 热键键定义整理集中管理，方便维护 *leexioua-keys-.ahk*, *leexioua-lbl-.ahk*
- 增加MACC支持，依赖Acc.ahk, Ankor.ahk 增加ADC新增任务的功能