

<http://www.autohotkey.com/>
<http://lukewarm.s101.xrea.com/>

windows 上の操作をマクロ化したり、キー操作を置き換えることが出来る。

AutoHotKey

スクリプトの実行の流れ

<https://sites.google.com/site/agkh6mze/howto/scripting>

1. スクリプトの読み込み
 1. 文法のチェック等
2. ディレクティブ評価
 - 1.IfWinActive など特殊なもの以外のディレクティブの評価
 - 2.Include、環境変数、エスケープ文字、スレッド関連など
3. 初回のスクリプト実行
 1. 先頭から順次実行される
 2. 以下のいずれかの条件で終了する
 1. スクリプトファイルの最後まで実行する
 2. ホットキーラベルや HotString の定義に遭遇
 - 3.Return や Exit のコマンドに遭遇
 - 4.ExitApp に遭遇
 5. 実行時エラーの発生
4. 常駐しないプログラムは終了
 1. 常駐の条件
 1. ホットキー、リマップ、HotString 定義がある
 - 2.GUI を利用している
 - 3.#Persistent で強制常駐モードにしている
 - 4.OnMessage() でメッセージ待ち状態にしている

様々な使い方

ウィンドウや GUI コントロールにウィンドウメッセージを送信する。

PostMessage を使うとウィンドウに直接ウィンドウメッセージを送信できる。

■ Rlogin.exe の次のウィンドウに移動

```
PostMessage 0x0111, 5068, , , A
```

0x0111 は メニュー操作やアクセラレータキー操作を通知

5068 は Rlogin.exe のメニュー ID

■ ウィンドウメッセージ

<http://note.phyllo.net/?eid=1106271>

■ メニュー ID (コマンド ID) を調べる

<http://ahkwiki.net/SendMessage>

XNResourceEditorPortable や Resource Hacker を使う。

https://portableapps.com/apps/utilities/xn_resource_editor_portable

<http://www.angusj.com/resourcehacker/>

調べたい実行ファイルを開き、ツリーの「Menu」以下を見ると、

```
MENUITEM "&File...  ¥tL", 40029
```

のような形式でメニュー項目が記述されている。
一番最後の数字が、メニュー項目に対応するコマンド ID である。

■ VirtuaWin でデスクトップ移動

```
PostMessage 1034, 1, 0, , ahk_class VirtuaWinMainClass
```

ウインドウメッセージとコマンド ID は VirtuaWin のヘルプに記載されている。

キーの送信方式を変える

```
SendMode
```

でキーの送信方式を変更できる。キーの動作がおかしい場合は送信方式を変えることで回避できることがある。

設定値	送信方法
Event	WindowAPI の <code>keybd_event</code> 、 <code>mouse_event</code> を利用して一つずつイベントを生成する方式。
Input	WindowsAPI の <code>SendInput</code> 関数を利用してシステムに一連の操作イベントをまとめて送り込む方式。他のプログラムがキーフックを利用していると性能が低下する。この方式が利用できない状況では、代わりに Event モードが使用される。
Play	<code>JournalPlayback</code> を利用して一連の操作イベントを再生させる方式。ゲームなどのプログラムをより確実に操作できる。
InputThenPlay	Input モードが利用できれば Input モード、利用できなければ Play モードになるようにする。

特定の send の場合だけ Input、Play にしたい場合は

```
sendInput  
sendPlay
```

を使うと良い。

sendplay は uac が有効な環境だとうまくいかない可能性あり

個人的には、Input だと一部のアプリで不具合があるので、Event を使う。
例えば、秀丸ファイラの Alt + U とか。

一時的にユーザの入力をブロックする

BlockInput

でユーザの入力をブロックできる。

設定値	効果
On	すべての入力を無効化する。
Off	入力を有効にする。
Send	Send や SendRaw コマンドを実行する際に自動的に入力ブロックが有効化されるようにする。(2000/XP 以外では無意味)
Mouse	MouseMove、MouseDown、MouseClickDrag コマンド実行時に自動的に入力ブロックが有効化されるようにする。(2000/XP 以外では無意味)
SendAndMouse	Send と Mouse を同時に設定。
Default	Send と Mouse の設定を OFF にする。
MouseMove	マウスカーソルを移動できなくする。
MouseMoveOff	BlockInput、MouseMove を解除する。

修飾キー以外のキーを修飾キーとして使う

修飾キーを増やすことは出来ないが、コンビネーションを使って修飾キーのように使うことが出来る。

```
vk1C & j::send {left}
```

ただし、2 つ以上のキーは制御できない。

```
+vk1C & j::send {left}  
!vk1C & j::send {left}
```

などの使い方もできない。

Windows の alt + @ をフックする。

Windows7 以降の IME は、日本語キーボードでも alt + @ を 半角 / 全角キーへ変換する。

alt + @ を他のアプリケーションで使いたいので、この機能が邪魔。

autohotkey を使って、alt + @ をフックして、アプリケーションへ渡す方法

■ 例 1 : アクティブなウィンドウに渡す

```
!vkC0::  
    ControlSend , , !{vkC0}, A  
    return
```

■ 例 2 : 例えば Java (SunAwtFrame) へ渡す方法

```
#IfWinActive ahk_class SunAwtFrame
!vkC0::
; send !{pgdn}
ControlSend , , !{vkC0}, ahk_class SunAwtFrame
return
#IfWinActive
```

Ctrl + ウィンドウズキー + U をフックする

Windows8 や Windows 2012 では

```
Ctrl + ウィンドウズキー + U
```

でナレータが起動してしまう。

例えば、このショートカットで Virtuawin の仮想デスクトップ 1 へ移動するには

```
#^U::
;run, "VirtuaWin.exe" -msg 1034 1
PostMessage 1034, 1, 0, , VirtuaWinMainClass
Return
```

とする。

IME の状態を取得する

を使う

XWindows のウィンドウには動作をさせない

```
GroupAdd, XWIN, ahk_class Xming X
GroupAdd, XWIN, ahk_class vcxsrv/x X r l
```

でグループを作成して

```
#IfWinNotActive ahk_group XWIN
なにか処理
#IfWinNotActive
```

のようにする。

処理をそのまま OS に受け流す

~ を使うと処理をそのまま OS に受け流しつつ処理を加えることが出来る

```
~^a::msgbox test
```

修飾子をそのまま受け流す

```
*b::send {blind}a
```

Capslock を Ctrl として使う

```
Capslock::Ctrl
sc03a::Ctrl
```

とする。ただ、Caps が特殊なためこの方法でも Ctrl UP が発行されず、Ctrl が押したままの状態となることが多発する。

自分で作ったスクリプト

■ 自分用キーバインド 1

```
suspend

;===== Include =====
#include lib\IME\UTF8\IME.AHK

;----- 表示ウインドウの設定 -----
Gui, +AlwaysOnTop -Caption
Gui, Font, s20 bold
Gui, Add, Text, cWhite vMode, MMM

;----- 初期化関数 -----
init:
    GroupAdd, XWIN, ahk_class Xming X
    GroupAdd, XWIN, ahk_class vcxsrv/x X r l
return

;**** リロード ****
; $!r::
; Reload
; return

; ファイル名編集 (Excel のセル編集) 等
!U::
    send {F2}
return

;-----
; ここからキー設定
;-----
; {Blind} を付けると修飾キーそのまま発行
; $*i::{Blind}{Up} など

; caps 以外にすれば動くと思う
; *capslock::
;     send,{Ctrl down}
;     return

; *capslock up::
;     send,{Ctrl up}
;     return

;**** 変換キー -> 半角全角
; $vk1C::send {vkF3}

; IfWinNotActive ahk_group XWIN
;**** 変換キー -> 半角全角
; $vk1C::
;     if(!IME_GET()){
;         send {vkF3}
;     }
;     return

;**** 無変換キー -> 半角全角
; $vk1D::
;     if(IME_GET()){
;         send {vkF3}
;     }
;     return
; IfWinNotActive

;***** サスペンド Toggle *****
; $!^u::
;     suspend
;     showSuspend()
;     return

; 無変換キーでオフ
```

```

$!vk1D::
    suspend, on
    showSuspend()
return

; 変換キーでオン
$!vk1C::
    suspend, off
    showSuspend()
return

$^vk1D::
    suspend, on
    showSuspend()
return

showSuspend(){
    ; suspend
    ; msgbox, %A_Issuspended%
    Gui, hide
    ;gui, show, x624 y998 NA

    if (A_Issuspended = 1) {
        GuiControl,, Mode, OFF
        Gui, Color, 2450DB
    } else {
        GuiControl,, Mode, ON
        Gui, Color, DB5D24
    }
    Gui, Show, AutoSize Center NA
    SetTimer, GuiHide, 500
    return
}

; ***** 画面を消す *****
GuiHide:
    SetTimer, GuiHide, off
    Gui, Hide
Return

;*** Ctrl+d ***
$^d::
    send {home}+{END}{space}{bs}{del}
return

;**** Alt + -> ESC
$!h::
    send {ESC}
return

:-----
:      カーソル移動
:      選択処理
:-----
:----- Alt -----
$!i::send {up}
$!k::send {down}
$!j::send {left}
$!l::send {right}
$!`::send {home}
$!vkBA::send {END}
; $!h::send {pgup}
; $!n::send {pgdn}
$!o::send {pgup}
$!.:send {pgdn}
$!0::send {del}
$!9::send {bs}

:----- Alt + Ctl -----
$^!i::send ^{up}
$^!k::send ^{down}
$^!j::send ^{left}
$^!l::send ^{right}
$^!`::send ^{home}
$^!vkBA::send ^{END}
$^!0::send +^{right}{space}{bs}
$^!9::send +^{left}{space}{bs}
$^!o::send ^{pgup}
$^!.:send ^{pgdn}

```

```

;----- Ctrl + Shift -----
;$^+i::send +{up}
;$^+k::send +{down}
;$^+j::send +{left}
;$^+l::send +{right}
;$^+`::send +{home}
;$^+vkBA::send +{END}
;$^+h::send +{pgup}
;$^+n::send +{pgdn}
;$^+o::send +{pgup}
;$^+.:send +{pgdn}

;----- Alt + Shift -----
;$!+i::send +{up}
;$!+k::send +{down}
;$!+j::send +{left}
;$!+l::send +{right}
;$!+`::send +{home}
;$!+vkBA::send +{END}
;$!+h::send +{pgup}
;$!+n::send +{pgdn}
;$!+o::send +{pgup}
;$!+.:send +{pgdn}

;----- Ctrl + Shift +Alt -----
;$^!+i::send ^+{up}
;$^!+k::send ^+{down}
;$^!+j::send ^+{left}
;$^!+l::send ^+{right}
;$^!+`::send ^+{home}
;$^!+vkBA::send ^+{END}
;$^!+o::send ^+{pgup}
;$^!+.:send ^+{pgdn}

```

■ 自分用キーバインド 2

```

; キーの動作が不安定な場合は SendMode を変更
; SendMode Input
; SendMode Play

;===== Include =====
#include lib\IME\UTF8\IME.AHK

;===== 初期化 =====
;----- X Server のウィンドウグループ
GroupAdd, XWIN, ahk_class Xming X
GroupAdd, XWIN, ahk_class vcxsrv/x X r l

;----- 表示ウィンドウの設定 -----
Gui, +AlwaysOnTop -Caption
Gui, Font, s20 bold
Gui, Add, Text, cWhite vMode, MMM

;----- 変数初期化 -----
isEnabled = 0

;----- 関数定義 -----
sendKey(key) {
; キー入力時の誤作動を防止
BlockInput On
; send %key%
sendInput %key%
BlockInput Off
}

; ===== ウィンドウ関連 =====
guimsg(msg, colour){
Gui, hide
GuiControl,, Mode, %msg%
Gui, Color, %colour%
Gui, Show, AutoSize Center NA
SetTimer, GuiHide, 500
}

GuiHide:
SetTimer, GuiHide, off
Gui, Hide

```

```

Return

; ===== キー操作 =====
; ----- リロード -----
; +^r::Reload

; ----- Windows 制御 -----
; # + u を無効
; *#u::return

; alt + @ による ime 制御を無効化
; !vkC0::
;   ControlSend , , !{vkC0}, A
;   return

; ----- CapsLock 関連 -----
; capslock
; (Ctrl up が発生しないため、Ctrl が押したままの状態になってしまうので注意
; Capslock::Ctrl
; sc03a::Ctrl

; ----- IME 関連 -----
; IME の設定で変換キー、無変換キーを設定している場合は必要なし
; #IfWinNotActive ahk_group XWIN
;   .**** 変換キー -> 半角全角
;   $vk1C::
;     if(!IME_GET()){
;       send {vkF3}
;     }
;     return
;   .**** カタカナ -> 半角全角
;   $vkF2::
;     if(!IME_GET()){
;       send {vkF3}
;     }
;     return
;   .**** 無変換キー -> 半角全角
;   $~vk1D::
;     if(IME_GET()){
;       send {vkF3}
;     }
;     return
; #IfWinNotActive

; ----- ツール毎の設定 -----
; Rlogin
; #IfWinActive ahk_exe RLogin.exe
;   . alt + @ の動作
;   !vkC0::
;     . WinGet, controls, ControlList
;     . StringSplit, classes, controls, `n
;     . msgbox, %classes6%
;     . ControlSend, %classes6%, !{vkC0}
;     . PostMessage 0x0111, 5068, , , A
;   return
; #IfWinActive

; VirtuaWin
; ctrl + win キー + U を VirtuaWinへ送信
; #^u::
;   PostMessage 1034, 1, 0, , VirtuaWinMainClass
;   Return

; ----- キー操作の切替 -----
; 無変換キーでオフ
; $!vk1D::
;   if (isEnable = 1) {
;     isEnable = 0
;     guimsg("OFF", "245DDB")
;   }
;   return

; 変換キーでオン
; $!vk1C::
;   if (isEnable = 0) {
;     isEnable = 1
;     guimsg("ON", "DB5D24")
;   }
;   return

```

```

; ===== キー操作 ( 切替可能 ) =====
#If (isEnable = 1)
; ファイル名編集 (Excel のセル編集) 等
$!u::sendKey("{F2}")
; $!d::sendKey("{home}+{END}{space}{bs}{del}")
$!h::sendKey("{ESC}")

; カーソル移動
$!i::sendKey("{up}")
$!k::sendKey("{down}")
$!j::sendKey("{left}")
$!l::sendKey("{right}")
$!`::sendKey("{home}")
$!vkBA::sendKey("{END}")
; $!h::sendKey("{pgup}")
; $!n::sendKey("{pgdn}")
$!o::sendKey("{pgup}")
$!::sendKey("{pgdn}")
$!0::sendKey("{del}")
$!9::sendKey("{bs}")

; ----- Alt + Ctl -----
$^!i::sendKey("^+{up}")
$^!k::sendKey("^+{down}")
$^!j::sendKey("^+{left}")
$^!l::sendKey("^+{right}")
$^!`::sendKey("^+{home}")
$^!vkBA::sendKey("^+{END}")
$^!0::sendKey("^+{right}{space}{bs}")
$^!9::sendKey("^+{left}{space}{bs}")

; ----- Ctrl + Shift -----
;$^+i::sendKey("+{up}")
;$^+k::sendKey("+{down}")
;$^+j::sendKey("+{left}")
;$^+l::sendKey("+{right}")
;$^+`::sendKey("+{home}")
;$^+vkBA::sendKey("+{END}")
;$^+h::sendKey("+{pgup}")
;$^+n::sendKey("+{pgdn}")
;$^+o::sendKey("+{pgup}")
;$^+::sendKey("+{pgdn}")

; ----- Alt + Shift -----
$!+i::sendKey("+{up}")
$!+k::sendKey("+{down}")
$!+j::sendKey("+{left}")
$!+l::sendKey("+{right}")
$!+`::sendKey("+{home}")
$!+vkBA::sendKey("+{END}")
$!+h::sendKey("+{pgup}")
$!+n::sendKey("+{pgdn}")
$!+o::sendKey("+{pgup}")
$!+::sendKey("+{pgdn}")

; ----- Ctrl + Shift +Alt -----
$^!+i::sendKey("^++{up}")
$^!+k::sendKey("^++{down}")
$^!+j::sendKey("^++{left}")
$^!+l::sendKey("^++{right}")
$^!+`::sendKey("^++{home}")
$^!+vkBA::sendKey("^++{END}")

```

#If