



Intro to AutoHotkey



AutoHotkey?

- An automation tool / scripting language
- Great for:
 - Automating repetitive tasks
 - Creating custom hotkeys / rebinding inputs
 - Hotstrings

```

; AutoHotkey
; Author: [unclear]
; Version: 1.0
; Description: [unclear]

; For internal use
; -----
; -----
; -----

Install()
{
    ; SilentMode := %S%
    ; AutoRestart := %R%
    ; DefaultType := %T%
    ; DefaultType := %A%
    ; DefaultType := %M%
    ; DefaultType := %D%
    ; DefaultPath := %P%
    ; DefaultPath := %I%
    ; DefaultPath := %O%
    ; DefaultPath := %U%
    ; DefaultPath := %L%
    ; DefaultPath := %S%

    Loop, %DefaultPath%
        Extract /%DefaultPath% %DefaultPath%

    ViewHelp %DefaultPath% /docs/scripts/HotInstall.htm
    ExitApp
}

; SilentMode := %S%
; AutoRestart := %R%
; DefaultType := %T%
; DefaultType := %A%
; DefaultType := %M%
; DefaultType := %D%
; DefaultPath := %P%
; DefaultPath := %I%
; DefaultPath := %O%
; DefaultPath := %U%
; DefaultPath := %L%
; DefaultPath := %S%

Loop, %DefaultPath%
    Extract /%DefaultPath% %DefaultPath%

ViewHelp %DefaultPath% /docs/scripts/HotInstall.htm
ExitApp

; SilentMode := %S%
; AutoRestart := %R%
; DefaultType := %T%
; DefaultType := %A%
; DefaultType := %M%
; DefaultType := %D%
; DefaultPath := %P%
; DefaultPath := %I%
; DefaultPath := %O%
; DefaultPath := %U%
; DefaultPath := %L%
; DefaultPath := %S%

Loop, %DefaultPath%
    Extract /%DefaultPath% %DefaultPath%

ViewHelp %DefaultPath% /docs/scripts/HotInstall.htm
ExitApp
; -----
; -----
; -----

; SilentMode := %S%
; AutoRestart := %R%
; DefaultType := %T%
; DefaultType := %A%
; DefaultType := %M%
; DefaultType := %D%
; DefaultPath := %P%
; DefaultPath := %I%
; DefaultPath := %O%
; DefaultPath := %U%
; DefaultPath := %L%
; DefaultPath := %S%

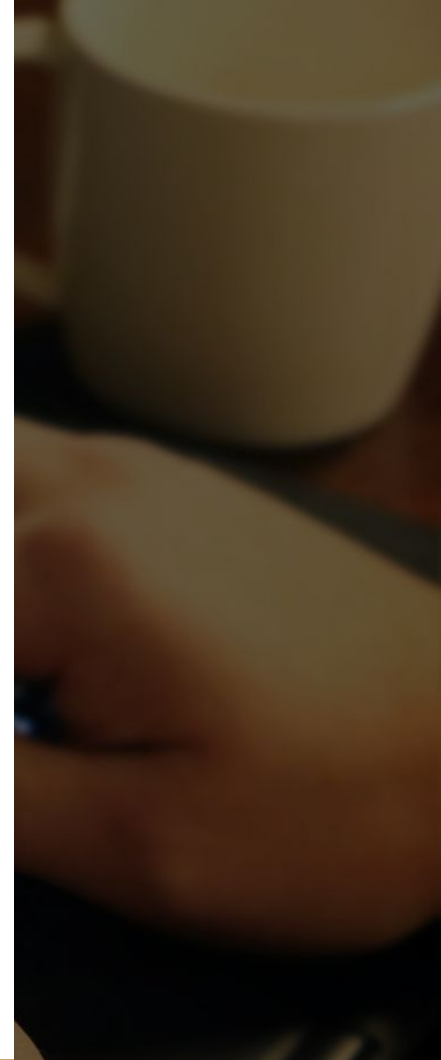
Loop, %DefaultPath%
    Extract /%DefaultPath% %DefaultPath%

ViewHelp %DefaultPath% /docs/scripts/HotInstall.htm
ExitApp
; -----
; -----
; -----

```

Automation is Great When Possible

- Execute tasks faster
- Multitasking
- More consistent
- Less Fatigue
- Scripts / tools are shareable



Automation with AutoHotkey

You can write scripts and bind them to hotkeys.

This code will open all the spreadsheets when control-1 is pressed

(^ means control)

(1 means 1)

```
OpenAllSpreadSheets ()
{
    OpenSpreadSheet ("TestData1.csv")
    OpenSpreadSheet ("TestData2.csv")
    OpenSpreadSheet ("TestData3.csv")
    OpenSpreadSheet ("TestData4.csv")
    OpenSpreadSheet ("TestData5.csv")
    OpenSpreadSheet ("TestData6.csv")
    OpenSpreadSheet ("TestData7.csv")
    OpenSpreadSheet ("TestData8.csv")
    OpenSpreadSheet ("TestData9.csv")
}

^1::
OpenAllSpreadSheets ()

return
```

Examples of Things I've Automated

- One press application testing
 - Run the application from visual studio
 - Log in as my development user
 - Navigate through menus, select options, wait for really long operations, etc...
- Batch import CSVs into Open Office
- Gather stock prices from the internet

Example: Batch Import CSVs

```
6
7   Ding()
8   {
9       SoundPlay, %A_WinDir%\Media\ding.wav
10  }
11
12  OpenSpreadSheet(filePath)
13  {
14      Run, "C:\Program Files (x86)\OpenOffice 4\program\scal" "%filePath%"
15
16      WinWaitActive, Text Import
17
18      sleep 100
19
20      MouseClick, Left, 594, 48
21
22      Ding()
23  }
24
```

```
24
25  OpenAllSpreadSheets ()
26  {
27      OpenSpreadSheet ("TestData1.csv")
28      OpenSpreadSheet ("TestData2.csv")
29      OpenSpreadSheet ("TestData3.csv")
30      OpenSpreadSheet ("TestData4.csv")
31      OpenSpreadSheet ("TestData5.csv")
32      OpenSpreadSheet ("TestData6.csv")
33      OpenSpreadSheet ("TestData7.csv")
34      OpenSpreadSheet ("TestData8.csv")
35      OpenSpreadSheet ("TestData9.csv")
36  }
37
38  ^1::
39  OpenAllSpreadSheets ()
40
41  return
42
```

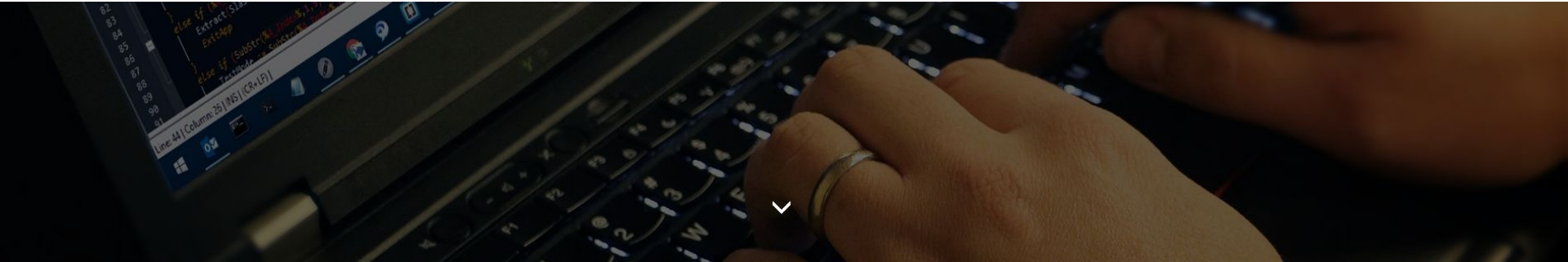
Example: Gathering Stock Prices

(let's look at the actual code for this one)

Key Rebinding

You can bind hotkeys to other inputs.

This is useful for customizing your workflow even if you're programs don't have customizable hotkeys.



Keybinding Examples

Mapping Keyboard Keys to the mouse wheel.

(Including horizontal mouse wheel)

```
^!Up::  
Send, {WheelUp}  
return  
  
^!Down::  
Send, {WheelDown}  
return  
  
^!Left::  
Send, {WheelLeft}  
return  
  
^!Right::  
Send, {WheelRight}  
return
```

Keybinding Examples

Left handed mouse friendly copy paste hotkeys.

```
^!,::  
Send, ^x  
return  
  
^!.::  
Send, ^c  
return  
  
^!/::  
Send, ^v  
return
```

HotStrings

- Type a short string and it will automatically replace it with another string
- Can also be used to execute scripts
- Quicker to customize than VS code snippets
- Usually requires less clicks / keypresses than VS Code snippets or auto complete

HotStrings: Examples

Replace "cst" with "c_str()":

```
::cst::c_str()
```

Insert debug stuff:

```
::pdo::__pragma(optimize("",off)) //HACK
```

Insert a static cast:

```
::stc::  
SendRaw, static_cast<>(  
Loop, 2  
{  
    Send, {left}  
}  
return
```

HotStrings: Examples

Insert a hack conditional breakpoint:

```
::cb-::  
SendRaw, static bool hackEnable = true; if (hackEnable && ) E_DEBUG_BREAK(); //HACK  
Loop, 25  
{  
    Send, {left}  
}  
return
```

HotStrings: Examples

Automatically create for loops:

```
::forx::  
SendRaw, for (size_t x = 0`; x < `; ++x)  
Loop, 6  
{  
    Send, {left}  
}  
return  
  
::fory::  
SendRaw, for (size_t y = 0`; y < `; ++y)  
Loop, 6  
{  
    Send, {left}  
}  
return  
  
::forz::  
SendRaw, for (size_t z = 0`; z < `; ++z)  
Loop, 6  
{  
    Send, {left}  
}  
return
```

Scripting Tips: Script Setup

I usually use these settings in my scripts:

```
#NoEnv ; Recommended for performance and compatibility with future AutoHotkey releases.
#Warn ; Enable warnings to assist with detecting common errors.
#SingleInstance force ; Let's me easilly re-run my script by clicking a short cut / double clicking script
SendMode Input ; Recommended for new scripts due to its superior speed and reliability.
SetWorkingDir %A_ScriptDir% ; Ensures a consistent starting directory.
```

Scripting Tips: Debugging

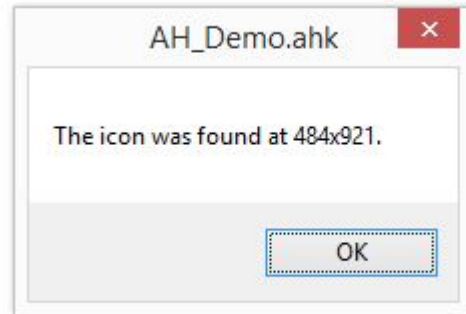
I don't *think* there's a debugger.

The MsgBox and ToolTip functions can come in handy to figure out what's going on in your scripts.

Scripting Tips: Debugging

MsgBox displays a pop up that blocks your script from executing until you click ok

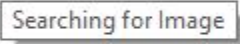
```
145  
146  
147  
MsgBox The icon was found at %FoundX%x%FoundY%.
```



Scripting Tips: Debugging

ToolTip displays text in a little tool tip box near your mouse without blocking.

```
140  
141  
142  
  
153  
154  
155  
  
ixel, Screen  
ouse, Screen  
  
)  
arch, FoundX, FoundY, 0, 0, A_ScreenWidt  
  
, Searching for Image
```

A screenshot of a script editor window. The editor shows several lines of code. A tooltip box is visible, containing the text "Searching for Image". The tooltip box is a small rectangle with a thin border and a light background, positioned near the right side of the code area. The code lines are numbered 140 through 155, with some lines being blank. The text "Searching for Image" is displayed in a monospaced font within the tooltip box.

Scripting Tips: Sound Notifications

- You can make scripts play sounds.
- This is nice to to notify you to pay attention after a long operation is done.

```
Ding()  
{  
    SoundPlay, %A_WinDir%\Media\ding.wav  
}
```

Scripting Tips: Waiting for Stuff

To automate things you often need to wait for things to happen before you execute the next step. Strategies include:

- Sleep function
- Various Built in Wait functions
- Loop with Check Function and Sleep function

Waiting for Stuff: Sleep

- Not very robust, but it's really quick and easy
- Lots of programs have little pauses, animations, etc. that are hard to check for.

```
100  
101  
102
```

```
sleep 100
```

Waiting for Stuff: Built in Wait functions

- These are usually ideal if one's available for what you need.
- Check the documentation to see what's available.

```
98  
99 WinWaitActive, Text Import  
100
```

```
201  
202 ClipWait  
203
```

```
410  
411 WinWaitActive, ahk_exe Program.exe  
412
```

Waiting for Stuff: Loop with a check in it

Good for unpredictable wait times when there isn't a built in wait function.

```
195 WaitForLargeWindow()
196 {
197     WinWaitActive, ahk_exe Program.exe
198
199     winX := 0
200     winY := 0
201     winW := 0
202     winH := 0
203     while (winW < 1900)
204     {
205         Sleep 1000
206
207         WinGetPos, winX, winY, winW, winH, ahk_exe Program.exe
208     }
209
210     Ding()
211 }
```

Scripting Tips: Image Search

- You can search the screen for images.
- Good for finding / clicking buttons, and more
- Note: the image needs to match what's on the screen fairly exactly
 - Watch out for button mouseover states, resolutions changes, etc.

Scripting Tips: Image Search

```
136
137 while (true)
138 {
139     ImageSearch, FoundX, FoundY, 0, 0, A_ScreenWidth, A_ScreenHeight, EnableCapture.bmp
140
141     Tooltip, Searching for Image
142
143     if (ErrorLevel = 0)
144     {
145         ;MsgBox The icon was found at %FoundX%x%FoundY%.
146
147         MouseClick, Left, %FoundX%, %FoundY%
148         break
149     }
150
151     Sleep 100
152 }
153
154 ToolTip ;Clearing the tool tip
155
```

Scripting Tips: Limiting Hotkeys to Certain Apps

- This is nice to help avoid accidentally executing scripts
- Also lets you re-use hotkeys for different things depending on what's open

```
561
```

```
562 #IfWinActive ProgramName - Microsoft Visual Studio
```

```
563
```

Scripting Tips: Predictable Coordinates

- You can change the coordinate mode for the mouse and more.
- Coordinates can be relative to things like the screen or active window.

```
134  
135 CoordMode Pixel, Screen  
136 CoordMode Mouse, Screen  
137
```

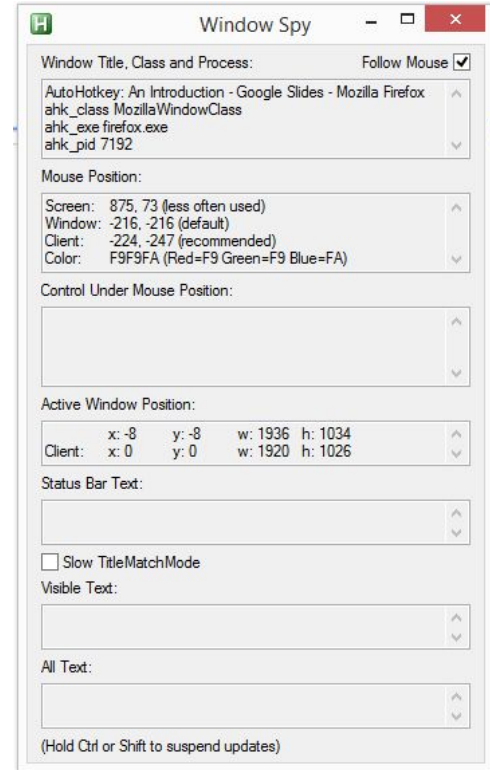
Scripting Tips: Predictable Coordinates

- You can also manipulate the active window to make things more predictable

```
601  
602 WinMove, 0, 0  
603 WinMaximize  
604
```

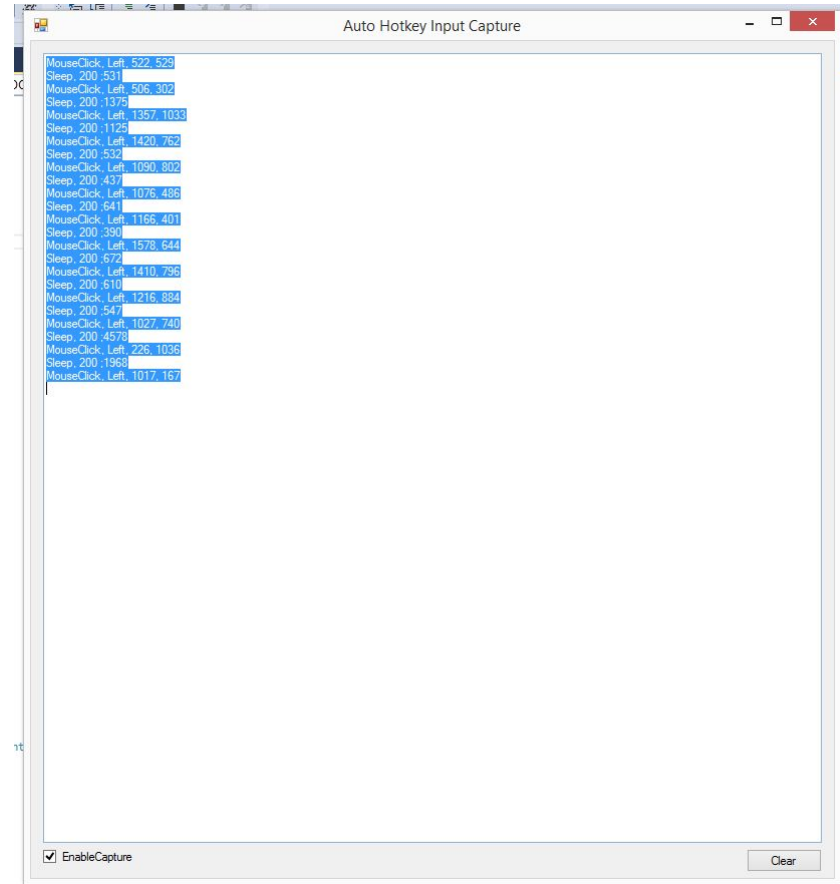
Cool Tools: Window Spy

- Comes with Autohotkey
- Useful for getting window information



Cool Tools: Input Recorder

- A quick script that records mouse inputs and timings.
- Outputs AutoHotkey code that can be copied into your scripts.
- Also has hacky limited screen capture for image based scripts.



Getting Started

- Go to: <https://www.autohotkey.com/>
- Download and install it.
- Check out the tutorial at: <https://www.autohotkey.com/docs/Tutorial.htm>

Questions?