

# What is working so far...

The working commands were tested with the script "autohotkeyce.ahk", included with the installer

**Green:** Command is working. Tested with the function "Test function"

**Red:** Command is not working. The reason is explained in the column Description

**Black:** Unknown/untested so far.

## Command List (from the ahk help)

Command	Description	Test function
{ ... }	A pair of braces denotes a block. Blocks are typically used with functions, Else, Loop, While-loop, and IF-commands.	
AutoTrim	Determines whether "Var1 = %Var2%" statements omit spaces and tabs from the beginning and end of Var2.	_autotrim()
BlockInput	Disables or enables the user's ability to interact with the computer via keyboard and mouse. <b>Minimum operating systems Windows 98, Windows 2000</b>	
Break	Exits (terminates) a loop. Valid inside any kind of loop.	_fileloop()
<u>ChangeDisplaySettings</u>	Changes the screen rotation	_ChangeDisplaySettings( )
Click	Clicks a mouse button at the specified coordinates. It can also hold down a mouse button, turn the mouse wheel, or move the mouse. <b>Click without parameters is not working because GetCursorPos (API) is not supported. So moving the mouse relative is not working, too because we can't get the current position of the cursor.</b>	
ClipWait	Waits until the clipboard	_Clipwait()

	contains data.	
Continue	Skips the rest of the current loop iteration and begins a new one. Valid inside any kind of loop.	_Continue()
<a href="#">Control</a> (ok)	Makes a variety of changes to a control.	
ControlClick	Sends a mouse button or mouse wheel event to a control. <b>Control-or-Pos:</b> xy, ClassNN, Text, HWND <b>WhichButton:</b> Left, Right (You can even use the right mouse button with ControlClick and catch it with OnMessage(0x204, "WM_RBUTTONDOWN") or GuiContextMenu:), (X1, X2, Middle, Wheelup, Wheeldown, WheelLeft, WheelRight not supported by OS) <b>Options:</b> NA D U Pos Xn Yn	_ControlClick()
ControlFocus	Sets input focus to a given control on a window.	_ControlFocus() _ControlSetText()
ControlGet	Retrieves various types of information about a control. <b>ControlGet,</b> Var, List, Selected   Colx   Count   Count Selected   Count Col, SysListView321 Checked, Enabled, Visible, Tab, FindString, Choice, LineCount, CurrentLine, CurrentCol, Line, Selected, Style, ExStyle, Hwnd	_Controlget() _Controlget1()
ControlGetFocus	Retrieves which control of the target window has input focus, if any.	_ControlGetFocus()
ControlGetPos	Retrieves the position and size of a control.	_ControlGetPos()
ControlGetText	Retrieves text from a control.	_ControlGetText()
ControlMove	Moves or resizes a control.	_ControlMove()
ControlSend / ControlSendRaw	Sends simulated keystrokes to a window or control.	_ControlSetText()
ControlSetText	Changes the text of a control.	_ControlSetText()
CoordMode	Sets coordinate mode for various commands to be relative to either the active window or the screen.	

	ToolTip, Pixel, Mouse, Caret, Menu	
Critical	Prevents the current thread from being interrupted by other threads.	_Critical()
DetectHiddenText	Determines whether invisible text in a window is "seen" for the purpose of finding the window. This affects commands such as IfWinExist and WinActivate.	_DetectHiddenText()
DetectHiddenWindows	Determines whether invisible windows are "seen" by the script.	_DetectHiddenWindow()
DllCall()	Calls a function inside a DLL, such as a standard Windows API function. <b>Thanks to fins for providing the code</b>	_DllCall()
Drive	Ejects/retracts the tray in a CD or DVD drive, or sets a drive's volume label. <b>Not supported by OS</b>	
DriveGet	Retrieves various types of information about the computer's drive(s). <b>List: There are no drive letters on this OS.</b> Capacity Filesystem, Label, Serial: <b>GetVolumeInformation not supported by the OS</b> Type: GetDriveType not supported by the OS Status <b>StatusCD: not supported by the OS</b>	_driveGet()
DriveSpaceFree	Retrieves the free disk space of a drive, in Megabytes.	_driveGet()
Edit	Opens the current script for editing in the associated editor.	_Edit()
Else	Specifies the command(s) to perform if an IF-statement evaluates to FALSE. When more than one command is present, enclose them in a block (braces).	_If()
EnvAdd	Sets a variable to the sum of	_env()

	itself plus the given value (can also add or subtract time from a date-time value). Synonymous with: var += value	
EnvDiv	Sets a variable to itself divided by the given value. Synonymous with: var /= value	_env()
EnvGet	Retrieves an environment variable. <b>Environment variables are not supported by the OS</b>	
EnvMult	Sets a variable to itself times the given value. Synonymous with: var *= value	_env()
EnvSet	Writes a value to a variable contained in the environment. <b>Environment variables are not supported by the OS</b>	
EnvSub	Sets a variable to itself minus the given value (can also compare date-time values). Synonymous with: var -= value	_env()
EnvUpdate	Notifies the OS and all running applications that environment variable(s) have changed. <b>Environment variables are not supported by the OS</b>	_env()
Exit	Exits the current thread or (if the script is not persistent contains no hotkeys) the entire script.	
ExitApp	Terminates the script unconditionally.	_ExitApp()
FileAppend	Writes text to the end of a file (first creating the file, if necessary).	_FileAppendDelete()
FileCopy	Copies one or more files.	_FileCopy()
FileCopyDir	Copies a folder along with all its sub-folders and files (similar to xcopy). <b>Not working yet. GetLastError=2 Requirements: OS Versions: Windows CE .NET 4.2 and later.</b>	_FileCopyDir()
FileCreateDir	Creates a folder.	_FileCreateDir()

FileCreateShortcut	Creates a shortcut (.lnk) file.	_FileCreateShortcut()
FileDelete	Deletes one or more files.	_FileAppendDelete()
FileInstall	Includes the specified file inside the compiled version of the script.	
FileGetAttrib	Reports whether a file or folder is read-only, hidden, etc.	_FileSetGetAttrib()
FileGetShortcut	Retrieves information about a shortcut (.lnk) file, such as its target file.	_FileCreateShortcut()
FileGetSize	Retrieves the size of a file.	_FileGetSize()
FileGetTime	Retrieves the datetime stamp of a file or folder.	_FileGetSetTime()
FileGetVersion	Retrieves the version of a file.	_FileGetVersion()
FileMove	Moves or renames one or more files.	_FileCopy()
FileMoveDir	Moves a folder along with all its sub-folders and files. It can also rename a folder.	_FolderCreateCopyDel()
FileRead	Reads a file's contents into a variable.	_FileRead()
FileReadLine	Reads the specified line from a file and stores the text in a variable.	_FileReadLine()
FileRecycle	Sends a file or directory to the recycle bin, if possible. The recycle bin is not supported by the OS	
FileRecycleEmpty	Empties the recycle bin. The recycle bin is not supported by the OS	
FileRemoveDir	Deletes a folder.	_FolderCreateCopyDel()
FileSelectFile	Displays a standard dialog that allows the user to open or save file(s).	_FileSelectFile()
FileSelectFolder	Displays a standard dialog that allows the user to select a folder. Not supported by Pocket PC 2003. I try to load function dynamically, so if the function exists on your device FileSelectFolder should work.	

FileSetAttrib	Changes the attributes of one or more files or folders. Wildcards are supported.	_FileSetGetAttrib()
FileSetTime	Changes the datetime stamp of one or more files or folders. Wildcards are supported.	_FileGetSetTime()
FormatTime	Transforms a YYYYMMDDHH24MISS timestamp into the specified date/time format.	_FormatTime()
GetKeyState	Checks if a keyboard key or mouse/joystick button is down or up. Also retrieves joystick status.	_GetKeyState()
Gosub	Jumps to the specified label and continues execution until Return is encountered.	_variables()
Goto	Jumps to the specified label and continues execution.	_variables()
GroupActivate	Activates the next window in a window group that was defined with GroupAdd.	_Group()
GroupAdd	Adds a window specification to a window group, creating the group if necessary.	_Group()
GroupClose	Closes the active window if it was just activated by GroupActivate or GroupDeactivate. It then activates the next window in the series. It can also close all windows in a group.	_Group()
GroupDeactivate	Similar to GroupActivate except activates the next window <b>not</b> in the group.	_Group()
GUI	Creates and manages windows and controls. Such windows can be used as data entry forms or custom user interfaces. Edit, Text ListBox ListView, Button Picture If you want to put controls on a tab control, please use Tab2	_GuiListBox() _GuiListBox() _guiListView() _GuiAddPicture()
<a href="#">GuiControl (Ok)</a>	Makes a variety of changes to a	

	control in a GUI window.	
GuiControlGet	Retrieves various types of information about a control in a GUI window. (blank) / Pos / Focus / FocusV / Enabled / Visible /Hwnd	_GuiControlGet()
Hotkey	Creates, modifies, enables, or disables a hotkey while the script is running.	_Hotkey()
if	Specifies the command(s) to perform if the comparison of a variable to a value evaluates to TRUE. When more than one command is present, enclose them in a block (braces).	_if()
if (expression)	Specifies the command(s) to perform if an expression evaluates to TRUE.	_if()
If var [not] between	Checks whether a variable's contents are numerically or alphabetically between two values (inclusive).	_if()
If var [not] in/contains MatchList	Checks whether a variable's contents match one of the items in a list.	_if()
If var is [not] type	Checks whether a variable's contents are numeric, uppercase, etc.	_if()
IfEqual/IfNotEqual	Compares a variable to a value for equality. Synonymous with: if var = value   if var <> value	_if()
IfExist / FileExist()	Checks for the existence of a file or folder.	_run() _funcs()
IfGreater/IfGreaterOrEqual	Compares a variable to a value. Synonymous with: if var > value   if var >= value	_if()
IfInString / InStr()	Checks if a variable contains the specified string.	_if()
IfLess/IfLessOrEqual	Compares a variable to a value. Synonymous with: if var < value   if var <= value	_if()
IfMsgBox	Checks which button was pushed by the user during the most recent MsgBox command.	_fileloop()

IfWinActive / IfWinNotActive	Checks if the specified window exists and is currently active (foremost).	_IfWinActive()
IfWinExist / IfWinNotExist	Checks if the specified window exists.	_gui()
ImageSearch	Searches a region of the screen for an image.	
IniDelete	Deletes a value from a standard format .ini file.	_INI()
IniRead	Reads a value from a standard format .ini file.	_INI()
IniWrite	Writes a value to a standard format .ini file.	_INI()
Input	Waits for the user to type a string (not supported on Windows 9x: it does nothing). <b>So it's not supported by WINCE</b>	
InputBox	Displays an input box to ask the user to enter a string.	_InputBox()
KeyHistory	Displays script info and a history of the most recent keystrokes and mouse clicks.	_KeyHistory()
KeyWait	Waits for a key or mouse/joystick button to be released or pressed down.	uncomment ;_KeyWait() ->
LeftClick	[Obsolete -- use Click for greater flexibility]	
LeftClickDrag	[Obsolete -- use MouseClickDrag for greater flexibility]	
ListHotkeys	Displays the hotkeys in use by the current script, whether their subroutines are currently running, and whether or not they use the keyboard or mouse hook.	_ListHotkeys()
ListLines	Displays the script lines most recently executed.	_ListLines()
ListVars	Displays the script's variables: their names and current contents.	_ListVars()
Loop (normal)	Perform a series of commands repeatedly: either the specified number of times or until break is encountered.	_loop()



Loop (files & folders)	Retrieves the specified files or folders, one at a time.	_FileLoop()
Loop (parse a string)	Retrieves substrings (fields) from a string, one at a time.	_LoopParse()
Loop (read file contents)	Retrieves the lines in a text file, one at a time (performs better than FileReadLine).	_LoopReadFile()
Loop (registry)	Retrieves the contents of the specified registry subkey, one item at a time.	_LoopRegistry()
Menu	Creates, deletes, modifies and displays menus and menu items. Changes the tray icon and its tooltip. Controls whether the main window of a compiled script can be opened.	_MenuTest()
MouseClicked	Clicks or holds down a mouse button, or turns the mouse wheel. NOTE: The Click command is generally more flexible and easier to use.	
MouseClickedDrag	Clicks and holds the specified mouse button, moves the mouse to the destination coordinates, then releases the button.	
MouseGetPos	Retrieves the current position of the mouse cursor, and optionally which window and control it is hovering over.	
MouseMove	Moves the mouse cursor.	
MsgBox	Displays the specified text in a small window containing one or more buttons (such as Yes and No).	_MsgBox()
OnExit	Specifies a subroutine to run automatically when the script exits.	_OnExit()
OnMessage()	Specifies a function to call automatically when the script receives the specified message.	_wm_LeftButtonDown()
OutputDebug	Sends a string to the debugger (if any) for display.	
Pause	Pauses the script's current thread.	_Pause()
PixelGetColor	Retrieves the color of the pixel at	_PixelGetColor()

	the specified x,y coordinates.	
PixelSearch	Searches a region of the screen for a pixel of the specified color. <b>GetDIBits does not exist on WinCE</b>	_PixelSearch()
PostMessage	Places a message in the message queue of a window or control.	
Process	Performs one of the following operations on a process: checks if it exists; changes its priority; closes it; waits for it to close. <b>Exist, Close, List, Priority (Windows CE does not support the concept of a priority class), Wait, Waitclose</b>	_Process()
Progress	Creates or updates a window containing a progress bar.	
Random	Generates a pseudo-random number.	_Random()
RegExMatch()	Determines whether a string contains a pattern (regular expression).	_RegExMatch()
RegExReplace()	Replaces occurrences of a pattern (regular expression) inside a string.	_RegExMatch()
RegDelete	Deletes a subkey or value from the registry.	_RegReadWrite()
RegRead	Reads a value from the registry.	_RegReadWrite()
RegWrite	Writes a value to the registry.	_RegReadWrite()
RegisterCallback()	Creates a machine-code address that when called, redirects the call to a function in the script.	
Reload	Replaces the currently running instance of the script with a new one.	_Reload()
Repeat...EndRepeat	[Obsolete -- use Loop for greater flexibility]	
Return	Returns from a subroutine to which execution had previously jumped via function-call, Gosub, Hotkey activation, GroupActivate, or other means.	
RightClick	[Obsolete -- use Click for greater	

	flexibility]	
RightClickDrag	[Obsolete -- use MouseClickedDrag for greater flexibility]	
Run	Runs an external program.	_Run()
RunAs	Specifies a set of user credentials to use for all subsequent uses of Run and RunWait. Requires Windows 2000/XP or later. <b>Not supported by OS</b>	
RunAt	Let the OS start an application at a specific time or deletes queued entries.	_RunAt()
RunWait	Runs an external program and waits until it finishes.	_RunWait()
Send / SendRaw / SendInput / SendPlay	Sends simulated keystrokes and mouse clicks to the active window.	_RotateScreenWithSend( ) _SendRaw()
SendMessage	Sends a message to a window or control and waits for acknowledgement.	_SendMessage()
SendMode	Makes Send synonymous with SendInput or SendPlay rather than the default (SendEvent). Also makes Click and MouseMove/Click/Drag use the specified method.	
SetBatchLines	Determines how fast a script will run (affects CPU utilization).	_SetBatchlines()
SetCapslockState	Sets the state of the Capslock key. Can also force the key to stay on or off.	_SetKeyboardState()
SetControlDelay	Sets the delay that will occur after each control-modifying command.	_ControlMove()
SetDefaultMouseSpeed	Sets the mouse speed that will be used if unspecified in Click and MouseMove/Click/Drag.	
SetEnv (Var = Value)	Assigns the specified value to a variable.	_Env()
SetFormat	Sets the format of integers and floating point numbers generated by math operations.	_SetFormat()

SetKeyDelay	Sets the delay that will occur after each keystroke sent by Send or ControlSend.	_SetKeyDelay()
SetMouseDelay	Sets the delay that will occur after each mouse movement or click.	
SetNumlockState	Sets the state of the Numlock key. Can also force the key to stay on or off.	_SetKeyboardState()
SetScrollLockState	Sets the state of the Scrolllock key. Can also force the key to stay on or off.	_SetKeyboardState()
SetStoreCapslockMode	Whether to restore the state of CapsLock after a Send.	
SetTimer	Causes a subroutine to be launched automatically and repeatedly at a specified time interval.	_SetTimer()
SetTitleMatchMode	Sets the matching behavior of the WinTitle parameter in commands such as WinWait.	_SetTitleMatchMode()
SetWinDelay	Sets the delay that will occur after each windowing command, such as WinActivate.	_SetWinDelay()
SetWorkingDir	Changes the script's current working directory. Current directory not supported by OS	
Shutdown	0 = Logoff 1 = Shutdown 2 = Reboot 4 = Force 8 = Power down	_Shutdown()
Sleep	Waits the specified amount of time before continuing.	
Sort	Arranges a variable's contents in alphabetical, numerical, or random order (optionally removing duplicates).	_GuiComboSort()
SoundBeep	Emits a tone from the PC speaker.	_Beep()
SoundGet	Retrieves various settings from a sound device (master mute, master volume, etc.)	

SoundGetWaveVolume	Retrieves the wave output volume from a sound device.	
SoundPlay	Plays a sound, video, or other supported file type.	_Beep()
SoundSet	Changes various settings of a sound device (master mute, master volume, etc.)	
SoundSetWaveVolume	Changes the wave output volume for a sound device.	
SplashImage	Creates or updates a window containing a JPG, GIF, or BMP image.	
SplashTextOn	Creates a customizable text popup window.	
SplashTextOff	Closes the above window.	
SplitPath	Separates a file name or URL into its name, directory, extension, and drive.	_SplitPath()
StatusBarGetText	Retrieves the text from a standard status bar control.	
StatusBarWait	Waits until a window's status bar contains the specified string.	
StringCaseSense	Determines whether string comparisons are case sensitive (default is "not case sensitive").	
StringGetPos / InStr()	Retrieves the position of the specified substring within a string.	_Strings()
StringLeft	Retrieves a number of characters from the left-hand side of a string.	_Strings()
StringLen / StrLen()	Retrieves the count of how many characters are in a string.	_Strings()
StringLower	Converts a string to lowercase.	_Strings()
StringMid / SubStr()	Retrieves one or more characters from the specified position in a string.	_Strings()
StringReplace	Replaces the specified substring with a new string.	_Strings()
StringRight	Retrieves a number of characters from the right-hand side of a string.	_Strings()

StringSplit	Separates a string into an array of substrings using the specified delimiters.	_Strings()
StringTrimLeft	Removes a number of characters from the left-hand side of a string.	_Strings()
StringTrimRight	Removes a number of characters from the right-hand side of a string.	_Strings()
StringUpper	Converts a string to uppercase.	_Strings()
Suspend	Disables or enables all or selected hotkeys and hotstrings.	_Suspend()
SysGet	Retrieves screen resolution, multi-monitor info, dimensions of system objects, and other system properties. MonitorCount, MonitorPrimary, Monitor, MonitorWorkArea, MonitorName	_SysGet()
Thread	Sets the priority or interruptibility of threads. It can also temporarily disable all timers.	
ToolTip	Creates an always-on-top window anywhere on the screen.	
Transform	Performs miscellaneous math functions, bitwise operations, and tasks such as ASCII/Unicode conversion.	
TrayTip	Creates a balloon message window near the tray icon. Requires Windows 2000/XP or later.	
UrlDownloadToFile	Downloads a file from the Internet.	_URLDownloadToFile()
Var = value	Assigns the specified value to a variable.	
Var := expression	Evaluates an expression and stores the result in a variable.	
VarSetCapacity()	Enlarges a variable's holding capacity or frees its memory. Normally, this is necessary only for unusual circumstances such as DllCall.	

While-loop	Performs a series of commands repeatedly until the specified expression evaluates to false.	
WinActivate	Activates the specified window (makes it foremost).	
WinActivateBottom	Same as WinActivate except that it activates the bottommost (least recently active) matching window rather than the topmost.	
WinClose	Closes the specified window.	_WindowFunctions()
WinGetActiveStats	Combines the functions of WinGetActiveTitle and WinGetPos into one command.	_WindowFunctions()
WinGetActiveTitle	Retrieves the title of the active window.	_WindowFunctions()
WinGetClass	Retrieves the specified window's class name.	_WindowFunctions()
WinGet	Retrieves the specified window's unique ID, process ID, process name, or a list of its controls. It can also retrieve a list of all windows matching the specified criteria.	_WindowFunctions()
WinGetPos	Retrieves the position and size of the specified window.	_WindowFunctions()
WinGetText	Retrieves the text from the specified window.	_WindowFunctions()
WinGetTitle	Retrieves the title of the specified window.	_WindowFunctions()
WinHide	Hides the specified window.	_WindowFunctions()
WinKill	Forces the specified window to close.	_WindowFunctions()
WinMaximize	Enlarges the specified window to its maximum size.	_WinMaximize()
WinMenuSelectItem	Invokes a menu item from the menu bar of the specified window.	
WinMinimize	Collapses the specified window into a button on the task bar.	
WinMinimizeAll	Minimizes all windows.	
WinMinimizeAllUndo	Reverses the effect of a previous WinMinimizeAll.	

WinMove	Changes the position and/or size of the specified window.	
WinRestore	Unminimizes or unmaximizes the specified window if it is minimized or maximized.	
WinSet	Makes a variety of changes to the specified window, such as "always on top" and transparency.	
WinSetTitle	Changes the title of the specified window.	
WinShow	Unhides the specified window.	
WinWait	Waits until the specified window exists.	
WinWaitActive	Waits until the specified window is active.	
WinWaitClose	Waits until the specified window does not exist.	
WinWaitNotActive	Waits until the specified window is not active.	
#AllowSameLineComments	Only for AutoIt v2 (.aut) scripts: Allows a comment to appear on the same line as a command.	
#ClipboardTimeout	Changes how long the script keeps trying to access the clipboard when the first attempt fails.	
#CommentFlag	Changes the script's comment symbol from semicolon to some other string.	
#ErrorStdOut	Sends any syntax error that prevents a script from launching to stdout rather than displaying a dialog.	
#EscapeChar	Changes the script's escape character (for example: backslash vs. accent).	
#HotkeyInterval	Along with #MaxHotkeysPerInterval, specifies the rate of hotkey activations beyond which a warning dialog will be displayed.	
#HotkeyModifierTimeout	Affects the behavior of hotkey	



	modifiers: CTRL, ALT, WIN, and SHIFT.	
#Hotstring	Changes hotstring options or ending characters.	
#IfWinActive / #IfWinExist	Creates context-sensitive hotkeys and hotstrings. Such hotkeys perform a different action (or none at all) depending on the type of window that is active or exists.	
#Include	Causes the script to behave as though the specified file's contents are present at this exact position.	
#InstallKeybdHook	Forces the unconditional installation of the keyboard hook.	
#InstallMouseHook	Forces the unconditional installation of the mouse hook.	
#KeyHistory	Sets the maximum number of keyboard and mouse events displayed by the KeyHistory window. You can set it to 0 to disable key history.	
#MaxHotkeysPerInterval	Along with #HotkeyInterval, specifies the rate of hotkey activations beyond which a warning dialog will be displayed.	
#MaxMem	Sets the maximum capacity of each variable to the specified number of megabytes.	
#MaxThreads	Sets the maximum number of simultaneous threads.	
#MaxThreadsBuffer	Causes some or all hotkeys to buffer rather than ignore keypresses when their #MaxThreadsPerHotkey limit has been reached.	
#MaxThreadsPerHotkey	Sets the maximum number of simultaneous threads per hotkey or hotstring.	
#NoEnv	Avoids checking empty variables to see if they are environment variables (recommended for all	

	new scripts).	
#NoTrayIcon	Disables the showing of a tray icon.	
#Persistent	Keeps a script permanently running (that is, until the user closes it or ExitApp is encountered).	
#SingleInstance	Determines whether a script is allowed to run again when it is already running.	
#UseHook	Forces the use of the hook to implement all or some keyboard hotkeys.	
#WinActivateForce	Skips the gentle method of activating a window and goes straight to the forceful method.	
#DisableTapAndHold	Special command for CE devices to disable tap and hold	

## Special things:

- AHKCE is supporting the **tap-and-hold** gesture. You have to create a gui. If you tap and hold you can see the animation (dots cearting a circle around the mouse cursor) and the label **GuiContextMenu** is automatically called.

Update 30-06-09: If using the API SHRecognizeGesture(&xxx) autohotkeyCE isn't working on PNA's any more.

Last Modified 21-10-10