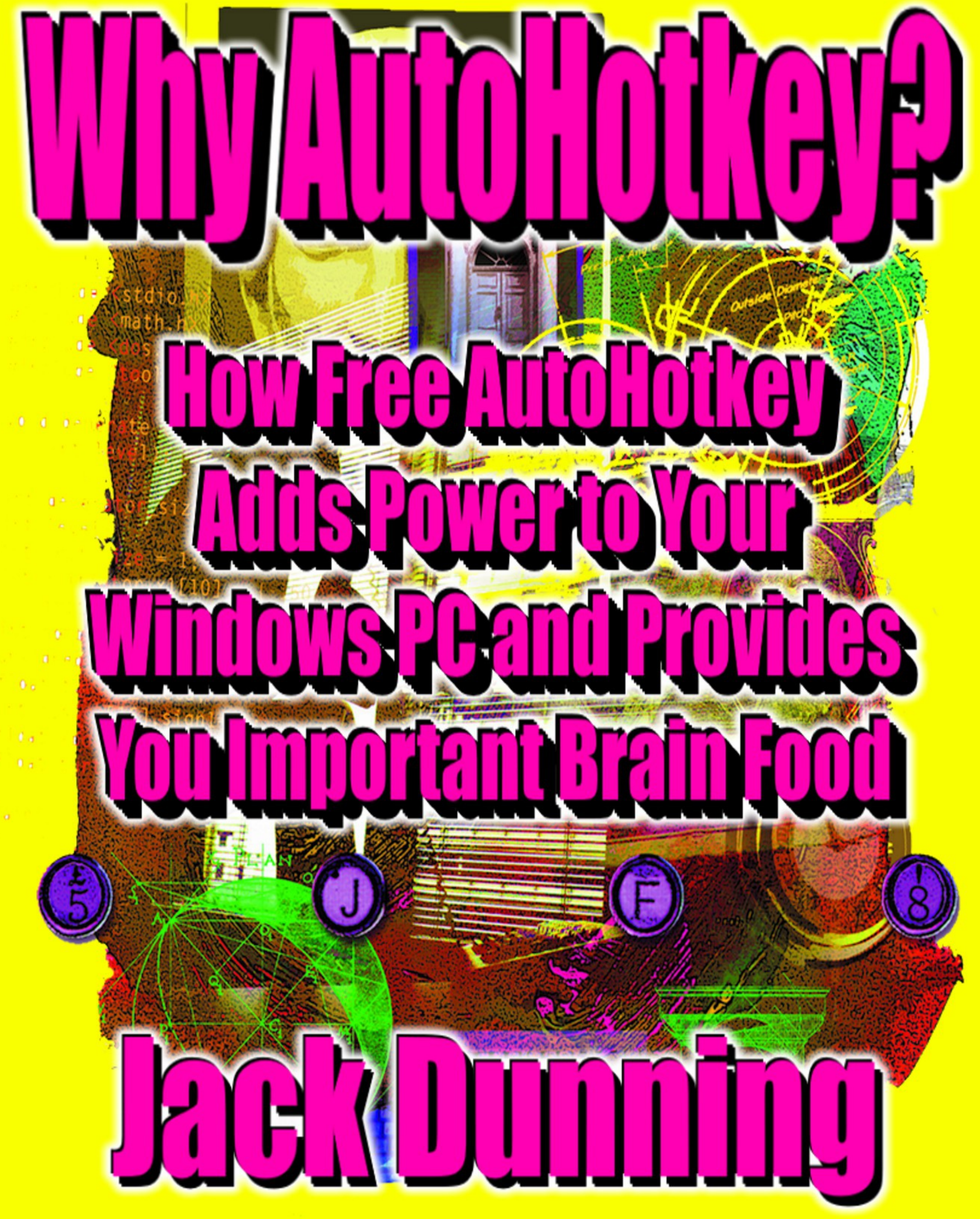
Why AutoHotkey?

How Free AutoHotkey Adds Power to Your Windows PC and Provides You Important Brain Food

Jack Dunning



Why AutoHotkey?

How Free AutoHotkey Adds Power to Your Windows PC and Provides You Important Brain Food

Ideas and Applications for How AutoHotkey Can Aid Various Occupations and Individuals in their Daily Computing Needs

by Jack Dunning

Table of Contents

Why AutoHotkey?
How Free AutoHotkey Adds Power to Your Windows PC and Provides You Important Brain Food
Ideas and Applications for How AutoHotkey Can Aid Various Occupations and Individuals in their Daily Computing Needs
by Jack Dunning
Why AutoHotkey for Every Windows Computer?
If, after reading this book, you don't come to the conclusion that you should make AutoHotkey part of your computing life, then you don't use a Windows computer. Maybe all you do is add one two-letter Hotstring which expands into your e-mail address. That alone will save you tons of time.
More AutoHotkey References
Why AutoHotkey for Learning to Program? Windows Programming Made Easy!
Never Programmed Before? You'll Find No Better Language for Learning Than AutoHotkey! Plus, It May Save Your Intellect from Precipitous Decline.
Languages for Learning to Program
AutoHotkey Gives Novices Instant Results
AutoHotkey Hotkeys
AutoHotkey Hotstrings for Text Expansion and Correction
Do As Little Or As Much As You Like
AutoHotkey Is an Easy-to-Read Language
Forgiving Structure and Syntax
Everything You Need Is on the Web…Mostly
Train Your Brain
Lifetime Benefits of Learning to Program
Why AutoHotkey for Writers, Bloggers, and Editors?
If You Write or Edit For a Living (or Fun) and Use a Windows Computer (Most People Do), Then You Should Use the Free AutoHotkey Software
AutoHotkey Works Everywhere
AutoHotkey Hotstrings
AutoCorrect—The Number One Reason for Writers to Use AutoHotkey
Free AutoHotkey Scripts and Apps
Scratching the Surface
Why AutoHotkey for Artists and Graphic Designers?
While Using AutoHotkey with Windows Makes Sense for Most Professions, Its Value Is Not So Obvious for Artists and Graphic Designers
Adding Enhancements to Your Graphic Tools
Empowering Windows Paint with AutoHotkey
Why AutoHotkey for Poets?
Erstwhile Multifarious Poets Optated for Quill and Parchment. Forthwith, AutoHotkey Propounds the Furtherance of Lyrical Ruminations on Windows Computers.
Beautiful Hotstrings
Rhymes from the Web
Why AutoHotkey for Chefs and Dieticians?
While You Don't Often See Television Chefs Using Computers, AutoHotkey Offers Guidance When Cooking "Jack Stuffed Cheeseburgers" at Home

Jack Stuffed Cheeseburgers
More Nutrition and Recipe AutoHotkey Scripts
The Egg Timer
Counting Calories
Recipe Book
Why AutoHotkey for Grandparents?
Remember All Your Grand Kid's Birthdays and Their Ages! There's No Limit to the Number of Ways You Can Amuse Your Grandchildren with AutoHotkey, Plus It Gives Your Brain a Much Needed Workout!
Remembering Birthdays and Ages
A Simple One-Line Script to Teach the Computer Keyboard and Alphabet
More AutoHotkey Entertainment and Education
Exercise the Brain
Why AutoHotkey for Students?
Ten Reasons Why Every Student Should Use the Free AutoHotkey Windows Tools! AutoHotkey Helps You So Much Academically That You'll Think You're Cheating! Plus, It's Delicious Brain Food and Helps Get You a Job!
Ten Reasons Students Should Use AutoHotkey
Reason One: AutoCorrect on the Fly the Most Commonly Misspelled English Language Words
Reason Two: Write Gender Neutral Essays and Term Papers
Reason Three: Insert Funky Foreign Language Characters
Reason Four: Instantly Add the Date to Any Document
Reason Five: Collect References for Research Papers
Reason Six: Protect Yourself from Lost Work
Reason Seven: Save Typing Time with Instant Hotkeys
Reason Eight: Eliminate Student's Most Commonly Overused Words in Papers
Reason Nine: Quickly Find Even More Synonyms
Reason Ten: Improve Your Mind and Get a Job
Why AutoHotkey for Teachers and Educators?
If You and Your Students Have Access to Windows Computers, Then You're Set! If Not, Well…
AutoCorrect That Teaches
The Web as a Reference
More Word Choice
Early Childhood Education
A Talking Keyboard
More Multimedia Presentations
Quiz Timing and Classroom Reminders
Teach Programming with AutoHotkey
Why AutoHotkey for Engineers and Scientists?
While Writing AutoHotkey Scripts Should Be No Problem for Most Engineers and Scientist, Many Might Be Surprised by How Much the Free Language Offers in Windows Tools
Scientific Terminology and Specialized Calculators
Inserting Mathematical and Scientific Symbols
AutoHotkey Hotstring Code
Special Purpose Calculators and Testers
Temperature Conversion Calculator
Creating the AutoHotkey Pop-up Window
Calculating Subroutines

Power Theory/Testing Tools
Why AutoHotkey for Internet Trolls?
If You Plan on Being One of the Most Annoying People on the Web, Why Not Make It Easy on Yourself?
Yell at Your Trolling Target
Don’t Use AutoHotkey AutoCorrect!
Disguise Yourself as a Candian
Canadian AutoHotkey AutoCorrect
E-Books for Making Your Journey into AutoHotkey Scripting Easier

Why AutoHotkey for Every Windows Computer?

I've written this book I call “Why AutoHotkey?” to illustrate the many reasons for Windows users to install and learn AutoHotkey. While I personally view AutoHotkey as "Absolutely the Best Free Windows Utility Software Ever!", millions of people around the world have no idea that the powerful scripting language even exists. The sole purpose of this book is to help the AutoHotkey unaware see how much the free software can aid them by adding power and flexibility to their Windows PC.

While most new AutoHotkey users start with setting up relatively simple Hotkeys and Hotstrings, this book gives a taste of some of the other easy techniques, as well as, a few of the more complex features of the scripting language. However, you don't need to go any further than Hotkeys for automating your Windows computer or Hotstrings for text expansion and autocorrection to get great value out of the free software.

Why AutoHotkey? is not a "how-to" book. There are plenty of other resources both in books and on the Web for that. Rather, I offer this book as a "why-to" which may stimulate you to discover how much easier AutoHotkey can make your computing life. Many of the examples offered here show and discuss some AutoHotkey code, but they are not intended as tutorials—although people comfortable with scripting will find it easy to pick up the language. (Tip for Beginners: Start with Hotkeys and Hotstrings. They are easy to add to your PC and give you instant results.)

This book only scratches the surface of what AutoHotkey can do. Many of its capabilities, such as Windows file and folder manipulation, don't appear in this book. The depth of the capabilities of this free Windows utility language makes it impossible to include many useful applications. However, with just a little investigation, you will find much more information on the Web sparking many additional ideas for how you can use AutoHotkey. I suggest my "Free AutoHotkey Scripts and Apps for Learning Script Writing and Generating Ideas" page and the scripts page on the main AutoHotkey site.

I've broken up *Why AutoHotkey?* by occupation. As examples and a way to show various applications for AutoHotkey, each chapter covers one or more field of endeavor where AutoHotkey can offer assistance. By no means a complete list of occupations or possible uses for AutoHotkey, the book offers insight into AutoHotkey techniques and apps which can improve your professional (and personal) Windows computing life. However, AutoHotkey is not limited to the script I highlight here. Far from it, providing the power of most other programming languages, the free Windows scripting language is only limited by your own imagination.

* * *

If you're new to AutoHotkey, see this “Introduction to AutoHotkey: A Review and Guide for Beginners.”

* * *

Note: A free scripting language for Window computers, AutoHotkey provides simple techniques for controlling and adding power to your PC. Since today (and into the foreseeable future) Windows PCs continue to represent 90% of the desktop and laptop computers, knowing how to write simple AutoHotkey scripts adds significant experience to anyone's résumé. Do it for fun and profit!

The following chapters demonstrate some of the ways I came up for using AutoHotkey:

• **Why AutoHotkey for Learning to Program? Windows Programming Made Easy**

Never programmed before? You'll find no better language for learning than AutoHotkey! Plus, it may save your intellect from precipitous decline.

• **Why AutoHotkey for Writers, Bloggers, and Editors?**

If you write or edit for a living (or fun) and use a Windows computer (most people do), then you should use the free AutoHotkey software.

Why AutoHotkey for Artists and Graphic Designers?

• While using AutoHotkey with Windows makes sense for most professions, it's not so obvious for artists and graphic designers.

• **Why AutoHotkey for Poets?**

Erstwhile multifarious poets optated for quill and parchment. Forthwith, AutoHotkey propounds the furtherance of lyrical ruminations on Windows computers.

• **Why AutoHotkey for Chefs and Dieticians?**

While you don't often see television chefs using computers, AutoHotkey offers guidance when cooking "Jack Stuffed Cheeseburgers" at home.

• **Why AutoHotkey for Grandparents?**

Remember all your grand kid's birthdays and their ages! There's no limit to the number of ways you can amuse your grandchildren with AutoHotkey. Plus, the free software gives your brain a much-needed workout!

• **Why AutoHotkey for Students?**

Ten reasons why every student should use the free AutoHotkey Windows tools! AutoHotkey helps you so much academically that you'll think you're cheating! Plus, it's delicious brain food and helps get you a job!

• **Why AutoHotkey for Teachers and Educators?**

If you and your students have access to Windows computers, then you're set! If not, well…

• **Why AutoHotkey for Engineers and Scientists?**

While Writing AutoHotkey scripts should be no problem for most engineers and scientist, many might be surprised by how much the free language offers in Windows tools.

• **Why AutoHotkey for Internet Trolls?**

If You plan on being one of the most annoying people on the web, why not make it easy on yourself?

Why AutoHotkey?
If, after reading this book, you don't come to the conclusion that you should make AutoHotkey part of your computing life, then you don't use a Windows computer. Maybe all you do is add one two-letter Hotstring which expands into your e-mail address. That alone will save you tons of time.

If, after reading this book, you don't come to the conclusion that you should make AutoHotkey part of your computing life, then you don't use a Windows computer. Maybe all you do is add one two-letter Hotstring which expands into your e-mail address. That alone will save you tons of time.

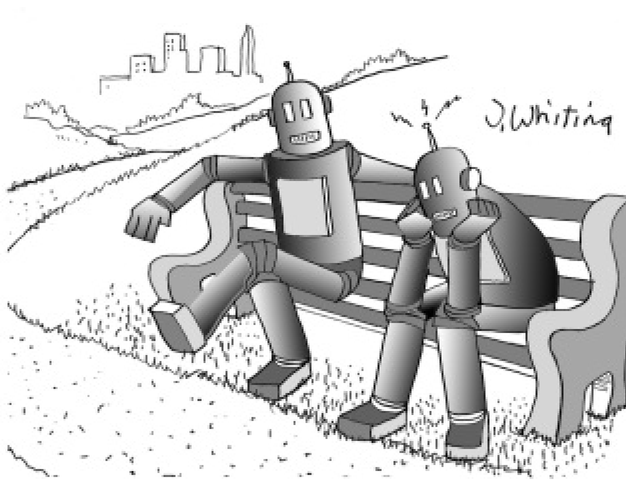
More AutoHotkey References

Why AutoHotkey for Learning to Program? Windows Programming Made Easy!

Never Programmed Before? You'll Find No Better Language for Learning Than AutoHotkey! Plus, It May Save Your Intellect from Precipitous Decline.

Languages for Learning to Program

I took a quick look at various sites on the Web recommending languages for learning how to program. You've got to be kidding me! I kept seeing lists similar to C, Java, C#, Perl, Ruby, and Python. These may be great recommendations for people who want to become professional programmers, but, if you merely want to make your personal computer more powerful or add useful features to PCs at work, these recommendations represent complex overkill. Anyone new to programming might find these languages incomprehensible—encountering a steep learning curve. Sure they'll be able to write the infamous beginning code for "Hello, World!", but the average person who has never coded before soon bogs down in enigmatic syntax (vocabulary) and structure (rules).



"Why does everyone assume that I know how to program?" Cartoon by Jim Whiting

Newcomers need easy software which teaches programming basics with each step—providing continuous satisfying results. That way the newbie stays motivated while developing relevant skills. This is where AutoHotkey excels as a beginning language for learning to program.

With AutoHotkey, you can achieve quick, simple, incremental success, yet, if you so desire, ultimately learn all of the concepts, techniques, and tricks valuable in any programming language. You don't need to take a great leap in understanding to get AutoHotkey working for you. The AutoHotkey learning process is naturally incremental—even without a tutorial. And yet, if you eventually want to create more complex apps, all the capabilities found in the more confusing languages used by the pros appear in AutoHotkey. Plus, if you want to make the jump to other languages later, the underlying principles you learn in AutoHotkey make that venture much easier.

You can start off as a neophyte who merely wants to control actions within your favorite program or game, then, if you continue with the natural process, end up understanding conditional *If* statements, *Loops*, and other common programming techniques—without even realizing your progress. The beauty of it is that you learn because you're doing something useful to you. Not like school. The only test is whether it works or not…and you'll know that from the first line of AutoHotkey you write.

AutoHotkey Gives Novices Instant Results

The lifeblood of a programmer's motivation is making the code work. When the coder eventually gets a routine running, there is an overwhelming desire to tell someone about the success. Sadly, few people understand enough programming to appreciate the brilliance of the achievement. (Too often programmers find themselves telling people about their cool, new routine only to be greeted with blank stares.) AutoHotkey not only provides quick results, but other non-computer people can appreciate the new features added to their Windows computer.

Unlike most other programming languages, AutoHotkey includes two programming structures which give results with as little as one line of code: Hotkeys and Hotstrings.

AutoHotkey Hotkeys

Hotkeys are key combinations which activate an action. For example, the key combination CTRL+L can be assigned to open a Google search page:

```
^l::run, http://google.com
```

This one line acts as a stand-alone app. Press the CTRL key and the L key simultaneously and Google search opens in your default Windows browser. (The caret ^ represents the Control key. The double-colon *::* makes it a Hotkey.) After saving this one line of coded in a text file with the AHK script extension, then loading it with the free open-source AutoHotkey program running on your Windows computer, the key combination immediately activates the Web page. (To get started with AutoHotkey, see "Installing AutoHotkey and Writing Your First Script.") Easy to test and easy to see results.

Suppose you want a shortcut for opening the Windows Calculator program:

```
^m::run, calc
```

After loading, CTRL+M (*^m*) launches the Windows Calculator (*run, calc*). Another one-line app.

Tip: If you have kids in the house, make it simple for them to launch their favorite "educational" program themselves by adding it to an AutoHotkey Hotkey script.

This only scratches the surface of what AutoHotkey can do with Hotkeys. Plus, these Hotkeys work in any Windows program or Web page. No other programming language makes producing useful Windows scripts so simple. But, wait! There's more!

AutoHotkey Hotstrings for Text Expansion and Correction

Another useful AutoHotkey technique similar to Hotkeys which proves just as simple while providing instant gratification includes Hotstrings. Similar in structure to Hotkeys, Hotstrings provide instant text manipulation and replacement:

```
::lol::laugh out loud
```

A double-colon *::* always precedes and follows the AutoHotkey activating Hotstring. The replacement text follows the second-double colon. After typing the activating string, followed by a space or punctuation, AutoHotkey deletes the activating string, substituting the replacement text. In this case, the one line of code (after loading the script on a Windows computer) replaces the typed characters “lol” (followed by a space or punctuation) with the text “laugh out loud” instantly in any document, editing window, or Web page.

Each line represents a stand-alone app. When building an autocorrect script, immediately correcting “teh” to “the” only requires the line:

```
::teh::the
```

(See “Add AutoCorrection to All Your Windows PC Programs with AutoHotkey” for many more autocorrect examples.) In most cases, AutoHotkey Hotkeys and Hotstrings can be added to the same file with little regard for their order.

Do As Little Or As Much As You Like

If you only use AutoHotkey to write Hotkey and Hotstring scripts, then you’ve done way more to add power to your Windows PC, than most people ever do. AutoHotkey includes most of the same capabilities for writing applications as any other programming language, but you don’t need to venture directly into those more complex areas—unless you get hungry for more. (For a taste of other possible AutoHotkey scripts and apps, see “Free AutoHotkey Scripts and Apps for Learning Script Writing and Generating Ideas.”)

You can learn as little or as much AutoHotkey as the motivation strikes you. However, in the course of writing books about AutoHotkey, I often found that each little piece I added to a script sparked new ideas for more cool features. That motivated me to move on to the next step—looking up more commands for adding that next bit. It’s easy to tinker with an old script (or someone else’s example) just to see how it reacts. I learned more from playing around with scripts than I have from reading about the commands online. True understanding of the nuances of a command comes from using it.

Maybe you’ll stop after adding a few useful Hotkeys/Hotstrings to your Windows computer. But if you keep adding to your scripts, you’ll learn more about programming than you ever thought possible. Rather than learning AutoHotkey all at once, you can grow into it. You can keep your apps simple or add as many features as you like. But, be careful! Too much tinkering may get you hooked.

AutoHotkey Is an Easy-to-Read Language

In some programming languages, the commands can look pretty cryptic. In AutoHotkey, most command and parameter names immediately suggest their function. "MsgBox" opens a message box. "Run" launches an app or opens a Web page. SetTimer starts a timer clock. This makes the commands easier to both understand and remember.

For experienced programmers this probably isn’t a significant point. When working with a new language, they know what type of command they want and have a pretty good idea where to look. But for the new or casual script writer, this type of easy reading commands may make the difference between getting a script running and giving up in frustration.

Forgiving Structure and Syntax

In most programming languages, the layout of a program file and writing of each command line must follow an exact format. If you don't get it right, nothing works. While not totally without its own formatting issues, AutoHotkey doesn't care about capitalization, the order of options in commands, and occasionally overlooks the dropping of a comma (e.g. the *MsgBox* command). This makes it a little easier for the newbie.

Note: AutoHotkey has a number of critical formatting issue in both script layout and command structure, but for the most part when starting out with Hotkeys and Hotstrings, formatting rarely present a problem (with the exception of the placement of double-colons).

Everything You Need Is on the Web…Mostly

Some people have a knack for computer coding. From the time they first look at a script, it immediately makes sense to them. They easily jump from one programming language to another. There might be a slight learning curve with the different vocabulary (systax) used in various languages, but that's just a matter of grasping the new jargon. It seems that their brain is networked for the logic of *If* statements and *Loops*.

For these computer savants, the Web provides everything they need through AutoHotkey documentation, forums, and help sites. They read the command definitions and immediately know what to do. They rarely need any extra help. (Admittedly, I have seen people operating at this high level pose questions on the AutoHotkey forums, but I often don't understand the question—much less the answer.) However, not everyone learns in the same way.

Since the my first look at AutoHotkey, I've attempted to make my writing a gentle introduction into the free Windows scripting language. Other than as a possible reference or idea generator, my books are not aimed at the professional programmer. The books are designed to aid the neophyte Windows script writer. If you're interested in taking a closer look at writing AutoHotkey scripts, but don't want to spend hours digging through the Web for the answers or don't feel totally comfortable with the idea of writing Windows scripts, then my book *A Beginner's Guide to AutoHotkey* might give you the start you want.

Train Your Brain

Sharper thinking may be the most important benefit from learning to write AutoHotkey scripts. In a world cluttered with insanity and arbitrariness, developing an understanding of how to write AutoHotkey scripts adds logic and discipline to a sloppy mind. Empowering your computer with AutoHotkey scripts changes the way your think. While you may not learn the answer to "Life, the Universe, and Everything", you will like the way AutoHotkey improves your mental acuity and eases your computing life. Plus, it's fun!

Many people think that programming is only for programmers. Not true! The beauty of AutoHotkey is that virtually anyone can learn to write simple one-line Windows enhancing Hotstrings and Hotkeys. If that were the only benefit a person received by using AutoHotkey, it would be plenty. But, as users learn more about AutoHotkey, bit by bit they venture into increasingly powerful tools. These new methods provide extra gains. In fact, even though you may start with simple scripts, the AutoHotkey language offers the depth of potential found in most programming languages—capable of producing full-featured desktop applications. Fortunately, the structure and implementation of AutoHotkey offers an ideal path for learning a little at a time.

Significantly, writing AutoHotkey scripts trains the mind in the analytical methods required in almost every phase of life. It teaches problems solving skills through the implementation of step-by-step logic paths and debugging techniques. The paramount importance of proper evaluation of the *If-Then-Else* decision model makes the analysis of any problem easier. The techniques for looping through series of alternatives gives users an understanding of the need to test all possibilities. Rather than coming out of school with a mishmash of contradictory thought processes, your mind graduates as a razor-sharp tool.

Lifetime Benefits of Learning to Program

While there have been a number of studies about what happens to the human brain in later years, most are inconclusive—especially when it comes to what to do about it. It seems that regular physical exercise helps, but the jury is out on how much various types of mental stimulation assist. However, most articles conclude that any form of cerebral gymnastics is "better than nothing" or "It can't hurt!"

From Time, "Games sure seem like a good way to work your brain out, but don't put your stock in Sudoku. 'They target very specific cognitive abilities, but they don't transfer to clarity of thinking, problem solving, planning—all the complex skills that really matter,' explains Dr. Sandra Bond Chapman, chief director of the Center for Brain Health at the University of Texas at Dallas and author of *Make Your Brain Smarter*."

"Along with aerobic exercise, engaging your brain in complex ways is absolutely necessary to keep your mind sharp in the second half of life." Studies have found few links between doing puzzles such as crossword and Sudoku and staving off mental deterioration. However, it may be that learning to write programs or computer scripts offers everybody increased cerebral longevity.

From the article, "Can Learning to Code Delay Alzheimer's?""

> "Several studies have shown that being bilingual or multilingual significantly delays the manifestation of Alzheimer's because bilingual brains have greater adaptability and functionality. So are programmers' brains similarly resistant to developing the disease?"

Professor Janet Siegmund of the University of Passau and her colleagues ran fMRI brain scans on 17 volunteers while they were reading code snippets for a study in 2014.

"We found [the] first empirical evidence that both, natural language and programming language, require the same areas in the brain," she said. "Based on this, we can infer that understanding programming languages and natural languages appear to be similar."

While not definitive, I've decided *not* to wait until science proves whether or not mental stimulation (and which type) actually helps keep my brain in shape. I'll continue doing crossword puzzles (and cheating via the Web) and exploring new ways to apply AutoHotkey on my Windows computer. I recommend that you do the same.

Why AutoHotkey for Writers, Bloggers, and Editors?

If You Write or Edit For a Living (or Fun) and Use a Windows Computer (Most People Do), Then You Should Use the Free AutoHotkey Software

Since I spend most of my time writing, it only makes sense that I start off with why wordsmiths should use AutoHotkey on their Windows computers. There exists a ton of tools for bloggers and editors which include built-in spell checkers and grammar checkers. AutoHotkey does not replace any of these but rather augments them with those extras which add an edge when writing. Best of all AutoHotkey works anywhere and everywhere on a Windows computer.

AutoHotkey, a free open source program, consists of a scripting language for adding universal features to any Windows computer. The scripts may be simple one-liners which create a Hotstring for text expansion and/or correction or more complex apps bestowing really cool features on your PC. You decide how much you want. Installing the free AutoHotkey software on your Windows computer puts numerous new capabilities within your reach.

AutoHotkey Works Everywhere

While many programs such as Microsoft Word include a number of writing tools, they only operate within that particular software. Once the writing environment changes, whether e-mail, Facebook, Web blogging software, or another Windows app, those extras in Microsoft Word no longer work. AutoHotkey breaks the barrier between Window programs by executing within any window—whether locally on the computer or remotely on the Web. This universality makes AutoHotkey a valuable Windows tool.

But how does AutoHotkey help the writer, blogger, or editor?

AutoHotkey Hotstrings

One of the features of AutoHotkey which distinguishes it from AutoIt, the original Windows automation language, includes a simple method for creating Hotstrings. AutoHotkey Hotstrings offer an easy way to execute text expansion and/or text correction. One line of code can perform an automatic real-time change to text when typing in any word processing program or Web editing field. For example:

```
::lol::laugh out loud
```

After loading this one line in an AutoHotkey script (*filename.ahk*), whenever you type “lol” followed by a space or punctuation, the text immediately replaces “lol” with the “laugh out loud” substitution phrase. AutoHotkey lets you create your own shorthand without forcing people to guess what you mean.

This technique expands commonly used abbreviations or adds special jargon to documents, but the available free AutoHotkey AutoCorrect app represents the most important way for writers to implement

Hotstrings.

"Who says I'm not qualified to be the editor?"

AutoCorrect—The Number One Reason for Writers to Use AutoHotkey

The AutoHotkey AutoCorrect script is not spelling or grammar correction software. There is plenty of other programs such as Grammarly which do that. The AutoCorrect.ahk script instantly fixes the most common spelling errors based upon thousands of Hotstrings created from Wikipedia's "Lists of common misspellings," "Typo," the Microsoft Office autocorrect list, the OpenOffice autocorrect list and various other sources. For example:

```
::mispelling::misspelling
```

Whenever you misspell the word by typing "mispelling", after hitting space or punctuation, it changes to the correct "misspelling" spelling.

The beauty of the AutoHotkey AutoCorrect app is that you can easily tailor it to your needs by editing and adding to the script. You can do that with any text editor (e.g. Notepad). Then, set up the script to load every time you boot into Windows by placing it in the Windows StartUp folder. The AutoCorrect app works everywhere, whether writing in a Word document, composing an e-mail, posting a Facebook comment, or working on a Web blog.

This AutoCorrect capability alone offers a strong incentive to investigate AutoHotkey.

But, there's more!

Free AutoHotkey Scripts and Apps

Over the last number of years, I've written (and stolen) quite a few AutoHotkey scripts. (Most of them I've posted for anyone to download and use in whatever way they wish.) The following list comprises a few of the AutoHotkey techniques and scripts which might prove most useful to writers, bloggers, and editors.

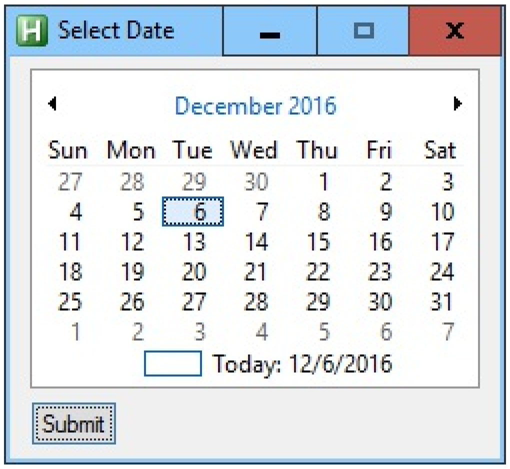
Quiet Down Those Shouting People by Changing Text to Lowercase

Surprisingly, I often run into the need to change the capitalization of text. Whether deliberate shouting (all capital letters) in a blog comment or needing to initial cap a headline, I use a few simple AutoHotkey Hotkeys which change any text to all lowercase, all uppercase, or initial cap each word in a phrase. The ChangeCase.ahk script includes all three.

After you select the target text by highlighting it, the Hotkeys secretly use the Windows Clipboard to change the case.

Add Today's (or Another Day's) Date to Your Writing

The script AddDate.ahk, includes a number of different ways to add today's date to any document. The string *Anow* (capital *A* required) instantly converts to the current date in US format (e.g. *December 6, 2016*). The string *Adate* (capital *A* required) opens the pop-up calendar shown for selecting any other date.



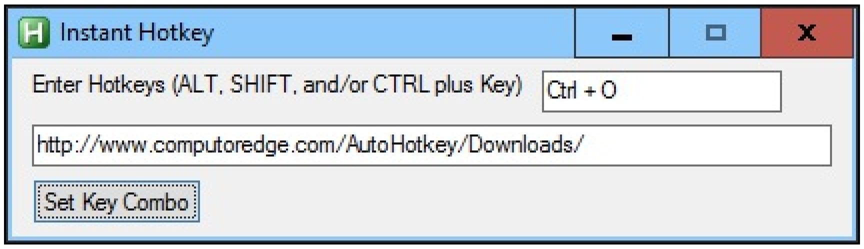
After entering *Adate* the calendar instantly pops up. Select the desired date and click *Submit*. AutoHotkey inserts the date into the document.

Use Google to Correct Common Phrases

I stole the clever GooglePhraseFix.ahk script. This little "Autocorrect Anything" by *aaston86* uses Google search to correct selected phrases by instantly accessing and returning the "Showing results for/Did you mean:" line in a Google search results page.

Instantly Add Temporary Hotkeys for Inserting Text

At times, I find myself using the same tedious text a number of times in the same article or blog. Who wants to type "www.computoredge.com/AutoHotkey/Downloads" over and over again. Rather than add it to the AutoCorrect file, I use my InstantHotkey.ahk script to create a temporary Hotkey for entering the text.



Ctrl + O

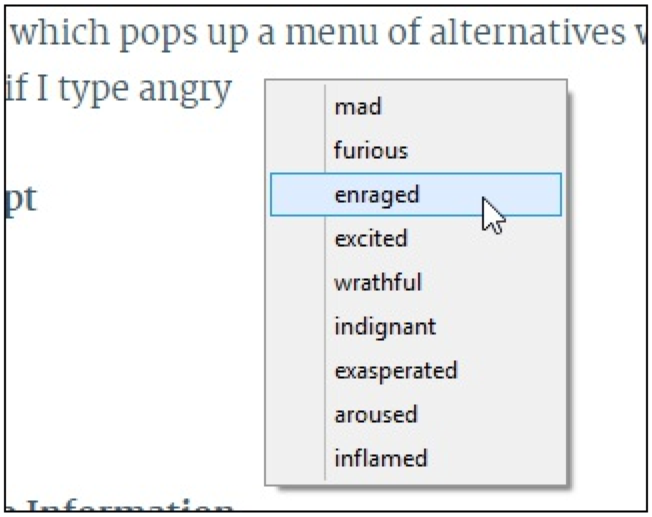
http://www.computoredge.com/AutoHotkey/Downloads/

After setting the Hotkey combination, whenever I press the keys simultaneously (CTRL+O for the image at left), AutoHotkey inserts the complete text (the URL shown). Whew! Now, I don't need to resort to copy-and-paste via the Windows Clipboard every time I need to enter that URL.

Tip: If you don't install AutoHotkey but still want to use some of the apps, then many of the scripts offer a compiled version (e.g. *InstantHotkey.exe*). If your Windows security settings allow you to download the compiled EXE file, then a simple double-click on the filename loads the app—main AutoHotkey installation not required. This comes in handy when using the script on another non-AutoHotkey PC or running the script from a thumb drive. (The EXE files are safe. I compiled each of them myself. Of course, that only works if you trust me. For extra safety, you can always compile the safe all-text AHK file into an EXE yourself.)

Redundant and Overused Words

While working on the book *Beginning AutoHotkey Hotstrings: A Practical Guide for Creative AutoCorrection, Text Expansion, and Text Replacement*, I came across a list of overused words compiled by high school English teachers. From that list, I wrote the script OverusedWords.ahk which pops up a menu of alternatives whenever one of the tired terms gets typed. For example, if I key in “angry”, the menu shown pops up. If I select “enraged”, it replaces “angry” in the text.



While the noobie AutoHotkey script writer may not know how to code this type of script on day one, it doesn’t take long to develop the skills needed to add many powerful features to a Windows computer.

Another Stolen Script

PhraseOMatic.ahk is an AutoHotkey script which makes it easier to enter key phrases into any document or text editing field for work and personal use. No more memorizing hotstrings or hotkey combinations. This script which puts all of your important phrases into a pop-up menu was written by Douglas Abernathy.

Quick Access to Web Information

Every good writer extensively uses references—now mostly found on the Web. AutoHotkey provides an easy way to speed up this Web access. I threw together a short script which opens *http://www.thesaurus.com* to find synonyms for a word highlighted in any Windows documents or Web page. The script I call SynonymLookup.ahk employs a standard AutoHotkey Windows Clipboard technique I discuss in most of my books:

```
^!l::
  OldClipboard:= ClipboardAll
  Clipboard:= ""
  Send, ^c ;copies selected text
  ClipWait 0
  If ErrorLevel
  {
    MsgBox, No Text Selected!
    Return
  }
  Run, http://www.thesaurus.com/browse/%Clipboard%
  Clipboard:= OldClipboard
Return
```

While this code may look complex to anyone unfamiliar with AutoHotkey, I can assure you that it’s easy to learn. Plus, even if you don’t learn each piece of this script, the only part you would need to change for extending this app to other Web sites would be the site URL shown in italics.

Scratching the Surface

The examples above are only a sampling of the AutoHotkey possibilities for writers, bloggers, and editors. For any wordsmith who decides to add new AutoHotkey features to their Windows computer, I recommend starting with the Hotstrings found in the AutoCorrect script. Then, as you familiarize yourself with the scripting language, add other useful features.

Why AutoHotkey for Artists and Graphic Designers?

While Using AutoHotkey with Windows Makes Sense for Most Professions, Its Value Is Not So Obvious for Artists and Graphic Designers

If you work in the graphic arts on computers, then you know that a multitude of programs exists for creating designs and original art pieces. Each one works a little differently with specific strengths and weaknesses. You may even use a couple of particular software packages for certain projects. Wouldn't it be nice to own a few tools which work in every one of those programs? AutoHotkey can do that!

"Jim was told that he could back up his data by making an image of his computer."

When producing ads and capturing screenshots, even I use a number of disparate apps for designing and finishing the artwork. (I have a preference for the free Paint.Net program.) However, no matter how powerful and feature-filled the software, I always want the tools to do a little more while working in each program. That's where AutoHotkey comes in.

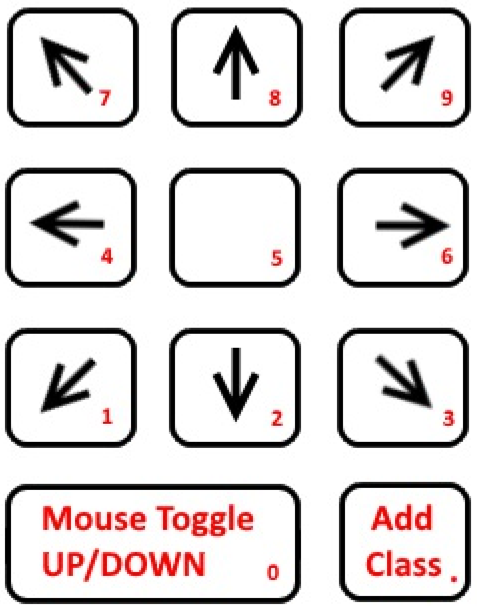
When AutoHotkey comes to mind, we first think of Hotkey shortcut actions and Hotstring text expansion and correction. While ideal for writers and bloggers, it's not obvious how AutoHotkey would work for the computer artist. But, if you have creative tendencies and use a Windows computer, then AutoHotkey can enhance the power of your PC graphics software. In addition to its unique program automation capabilities, AutoHotkey offers commands which make many cursor and mouse movement tasks easier:

- The Click command (and MouseClick command) make it possible to precisely position the cursor at any location on the screen or within a window. The *MouseClick* command has similar functionality to the *Click* command, but "the *Click* command is generally more flexible and easier to use."
- The MouseMove command (and Send command) facilitate the automatic movement of the cursor to a new location.
- The MouseClickDrag command holds down the specified mouse button while moving the cursor to a new location—then releasing the button.
- The PixelGetColor command and PixelSearch command copies colors at a screen location and finds specific colors within a window area respectively.
- ImageSearch command locates specific images within an area. (Often used to identify icons and buttons, I haven't yet played with *ImageSearch* command.)

- The BlockInput command allows the turning off of the mouse (or keyboard) while other actions take place.

Integrate these commands with any Windows software to build super tools. For example, I wrote an AutoHotkey script called MousePrecise.ahk which turns the keyboard numeric keypad into a mouse controller for moving the cursor one pixel at a time in one of eight directions. It works in any Windows program. (Find detailed discussions of the *MousePrecise* script beginning at my AutoHotkey blog and in the book *AutoHotkey Hotkey Techniques*.)

Note: Because script writing looks mysterious to many people, the myth spreads that only programmers can do it. Not true! Once the tools are understood, writing AutoHotkey scripts turns into very creative process ripe for anyone with the ability to solve puzzles. I contend that artists make some of the best script writers in the world.



In the *MousePrecise* script, pressing the 0 (zero) key holds the left mouse button down while other numbers become micromovement cursor keys—one pixel at a time. Press 0 again to release the left mouse button. Press Add Class (.) to activate the feature for a new active program.

Adding Enhancements to Your Graphic Tools

Rather than acting as a replacement program for specialized graphic software, AutoHotkey works best for creating tool add-ons which work in any Windows software. While the built-in tools in any given program may be great, there's always something which makes them operate a little better. AutoHotkey does that.

AutoHotkey does not change the nature of any built-in graphic tools. It merely adds features taking any Windows program to the next level. As an example, I used the Window Paint program to demonstrate how AutoHotkey might improve the built-in tools.

Empowering Windows Paint with AutoHotkey

Since Windows Paint is such a simple program, it makes a great example. In a series of columns I wrote in the beginning of 2015 for *ComputorEdge Magazine*, I highlighted how AutoHotkey can control and add capabilities to Windows Paint tools. These AutoHotkey techniques can work with other Windows graphic software:

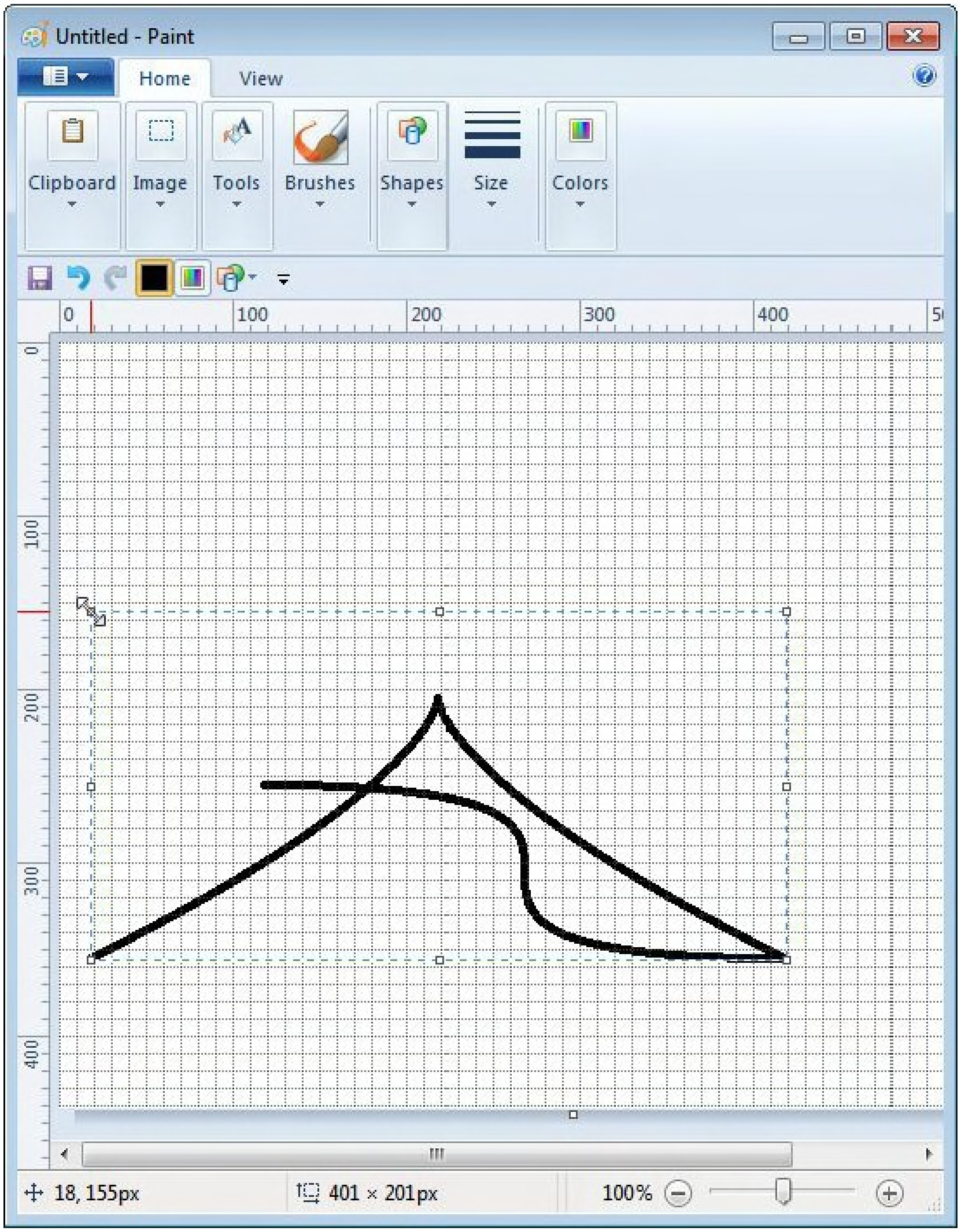
- "Controlling Windows Programs with AutoHotkey"—Windows Paint is used to demonstrate how to draw and fill in a square with AutoHotkey. Find tips which help you to both automate the drawing of simple objects and control other types of applications with AutoHotkey.
- "Using AutoHotkey to Draw a Straight Line in Windows Paint"—While you may never need to do this, these AutoHotkey Techniques apply to many other applications. Continued investigations into controlling AutoHotkey drawing tools in Windows Paint.
- "Make Your Own Drawing Tools for Windows Paint with AutoHotkey"—Build functions with

AutoHotkey to create controls for Windows Paint (or any other program). Learn how to use the same line drawing code over and over again by putting it in an AutoHotkey function. Then draw a line of any length at any angle anywhere in Windows Paint using only one line of code.

- "Automatically Find Buttons by Color in Windows Programs"—Yet, one more reason to use AutoHotkey free software! This little know AutoHotkey command will scan a section of a window or image for a specific color. Use it to select colors in Windows Paint.
- "Controlling More Windows Paint Tools with AutoHotkey"—Make identical squiggles, another reason to use AutoHotkey free software! One more step taken toward building AutoHotkey tools for controlling Windows programs.

Although not necessarily beginning level articles, these discussions offer insight into whether AutoHotkey might help you decide as a graphic artist whether or not you can improve your current Windows software by adding extras. (For your convenience, I extracted these columns from the original *ComputorEdge* issues and combined them into one free PDF download: AutoHotkey for Artist.pdf (available at *http://www.computoredge.com/AutoHotkey/Downloads/AutoHotkey%20for%20Artists.pdf*).)

As an example, short AutoHotkey scripts can replicate mouse movements accurately without human error. Discussed in the article "Controlling More Windows Paint Tools with AutoHotkey" a function uses the Paint Curve tool to create the figure shown in the screenshot below.



This script eliminates the human error cause by hand manipulation of a mouse. With each script activation, AutoHotkey replicates the same action. Change the script parameters to alter the result.

While the scripts listed in these articles demonstrate many graphic tool possibilities, my favorite artwork oriented AutoHotkey app has to be the MousePrecise script discussed above. I use it whenever I need to accurately place and/or drag the mouse cursor. I'm sure that many graphic programs offer similar features, but this one works in any Windows program.

If you’re a computer artist or graphic designer, you have plenty of reason to investigate AutoHotkey. As with any creative project, only your imagination limits AutoHotkey.

Why AutoHotkey for Poets?

Erstwhile Multifarious Poets Optated for Quill and Parchment. Forthwith, AutoHotkey Propounds the Furtherance of Lyrical Ruminations on Windows Computers.

Okay…I'm not a poet. My mind doesn't work that way. But that doesn't mean I can't see how AutoHotkey might be useful to people who craft the English (or any other) language. Even so, I occasionally enjoy writing a short rhyming couplet. (I know…constructing rhyming poems has become cliché—at least for real poets.)

In this blog, I offer a couple of AutoHotkey scripts for assisting and inspiring(?) budding wordsmiths. The first includes a set of over 500 Hotstrings for inserting "the most beautiful words in the English language." The second script draws upon the Web to create a pop-up menu of rhymes. Even if you never intend to write a poem, you might find these AutoHotkey techniques interesting and/or useful.

Beautiful Hotstrings

After discovering a list of the "117 most beautiful words in the English language" I put together a script with over 500 Hotstring replacements. I thought that poets might want to automatically replace some of the more mundane words with eloquent substitutions. After all, poetry is the art of bringing out the elegance of words in both speech and writing. Upon loading these Hotstring, AutoHotkey replaces any word found in the common list with the theoretically more beautiful alternative. A sample of the Hotstrings follows:

```
::secretive::surreptitious
::hidden::surreptitious
::secret::surreptitious
::undercover::surreptitious
::underground::surreptitious
::whispering::susurrous
::hissing::susurrous
::murmuring::susurrous
::locket::talisman
::amulet::talisman
::spell::talisman
::tinkling::tintinnabulation
::ringing::tintinnabulation
```

```
::clinking::tintinnabulation
::cover::umbrella
```

Each Hotstring line of code in this PoeticWords.ahk script acts as an independent command. Once (*erstwhile?*) loaded, whenever you type the first word (located between the two sets of double colons) followed by keying any standard punctuation or space, AutoHotkey deletes that word and replaces it with the remaining text (located after the second set of double colons). To add more Hotkeys, simply insert another line using the same format.

Warning (*harbinger?*): Don't leave these AutoHotkey Hotstrings loaded when writing comments on Facebook…unless you want to appear pretentious.

The AutoHotkey Hotstring feature is an important (*quintessential?*) technique for creating writing and editing apps. One popular AutoHotkey script corrects (on the fly) commonly misspelled words in any Windows document or editing field. You can adapt these text expansion/replacement capabilities in any manner that suits your personal needs. In the book *Beginning AutoHotkey Hotstrings*, I offer examples for automatically expanding abbreviations into their full definition, as well as, show how to insert boilerplate text. I even wrote a short script for popping up a menu of substitutes for overused words found in school term papers.

This PoeticWords.ahk script works best as a tool for alternative word exploration rather than an app for always loading on your PC. The "beautiful" replacement words may pop-up at unexpected times. Plus, the meaning of the new (*incipient?*) words may not quite coincide with the original term. However, the surprising substitute may inspire further digging into its appropriateness. If the alternative doesn't work, then in most Windows programs pressing CTRL+Z removes it. (Click the mouse button before finishing the word to disable any Hotstring about to activate.) If the script becomes an annoyance, simply right-click on its icon in the Windows System Tray and click *Exit*.

The key for poets is adapting the Windows AutoHotkey tools for creative purposes. I left out a few of the "beautiful word" suggestions since I couldn't see any obvious replacements. I also added a few of my own. No doubt any poet would want to include their own favorites. (Okay, maybe you don't need to write a poem, but you can always make your messages and e-mails more ostentatious by running this app.)

If I taught a class in creative poetry (as if I would qualify), I might write an app similar to the overused words script mentioned above for suggesting "beautiful words" to students while they type. It would take some time to assemble the word lists, but the script could then be compiled into an EXE file to run on any Windows computer. Then, the student could use the app without even loading the main AutoHotkey program.

Rhymes from the Web

In these times, I don't know how many (*a plethora*?) poets look for rhyming words. It seems that, with the exception (*elision?*) of limericks, rhymes may have gone out of fashion. However, for beginning (*incipient?*) lyricists, a method for locating similar sounding words might make the job a little easier.

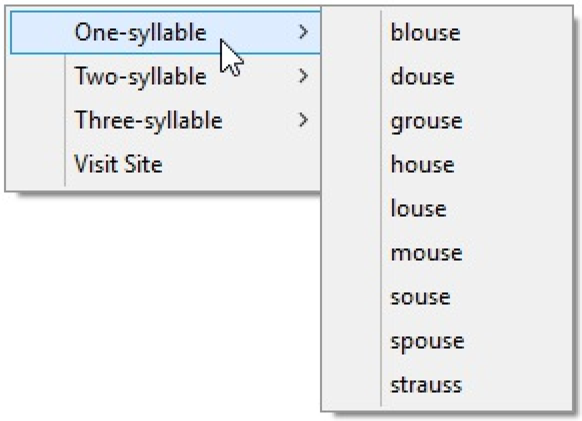
The Web acts as a tremendous source of free information. Normally, we access it by loading a Web browser and navigating to (or searching for) a specific page. A number of Web pages offer rhymes. With AutoHotkey, we can bypass the Web browser and directly load the alternative rhymes from the Web page into a pop-up menu. Select the word from the menu, then AutoHotkey inserts it into the document.

This RhymeMenu.ahk script helped me write the following poem:

```
As I sat in my house,
    Looking at a mouse,
I found that the louse
    Was eating my blouse!
```

(I don't actually own a blouse, but I took poetic license. After hearing the poem, my granddaughter smiled, "That's silly!")

Admittedly, RhymeMenu.ahk is not a beginning (*incipient?*) level script. Therefore, I do not offer a detailed discussion of the code here. However, I have embedded comments in the AHK file which highlight how it works.



After highlighting the word "house" and using the Hotkey combination CTRL+ALT+R, a menu pops up displaying a list of possible rhymes. Click a word and it replaces the originally highlighted word.

Briefly, the script uses the Windows Clipboard to capture the selected word and downloads the source code from the rhyming page (*http://www.rhymer.com/RhymingDictionary/*)— using the captured word as a search term. Then, calling upon the mysterious (*enigmatic?*) Regular Expression functions, AutoHotkey parses the Web page source code to pull out the lists of rhymes. Next, the AutoHotkey script inserts those rhyming word lists into menus for both display (*panoply?*) and action. Pretty (*fetching?*) cool, huh? But, not for the faint (*evanescent?*) of heart. (While not for beginners, I discuss the enigmatic Regular Expression in depth in the book, *A Beginner's Guide to Using Regular Expressions in AutoHotkey*.)

The primary problem I encountered with this script occurred whenever the Web page included an ad in the middle of a list of rhymes. It didn't properly parse the word list for the menu. In those cases, the word "None" appears as the only choice…and does nothing. For that reason, I added to the main menu the option to load the Web page (*Visit Site*) with your default browser.

These two scripts represent only a couple of the AutoHotkey possibilities for budding poets. Once (*erstwhile?*) you add AutoHotkey as a second language (ASL) to your vocabulary, you can steep yourself in Windows scripting creativity.

Note: You may have noticed a number of places in this chapter where a word in parentheses and terminated with a question mark, e.g. *(incipient?)*, follows another fairly common term. That enclosed word sprung from the *PoeticWords.ahk* script which I maintained loaded while (*erstwhile?*) writing this piece. Many (*a plethora?*) were not appropriate or didn't fit my meaning, but it sure got me thinking.

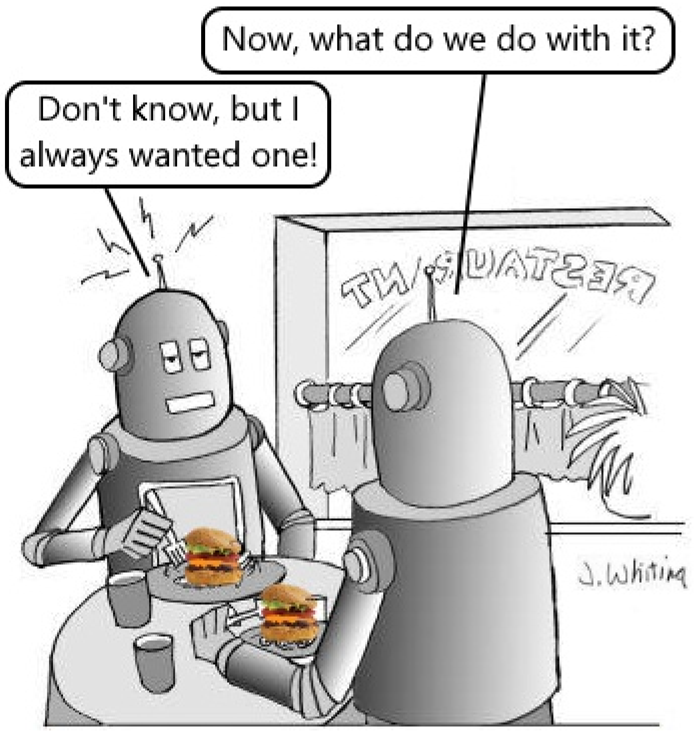
Why AutoHotkey for Chefs and Dieticians?

While You Don't Often See Television Chefs Using Computers, AutoHotkey Offers Guidance When Cooking "Jack Stuffed Cheeseburgers" at Home

* * *

While I came up with a pretty cool barebones recipe script for this chapter, I'm not sure how well computers and cooking mix in the kitchen. From what I've seen, professional chefs don't have much time to mess with technology—except for possibly writing cookbooks—in which case, I refer them to "Why AutoHotkey for Writers, Bloggers, and Editors?"

* * *


Now, what do we do with it?
Don't know, but I always wanted one!
RESTAURANT
J. Whiting

Using a computer while cooking presents a challenging task. Although looking up a recipe on a smartphone works great, poking at the device with fingers covered in cookie dough proves impractical. Ideally, any kitchen machine is lightweight and stands upright on its own.

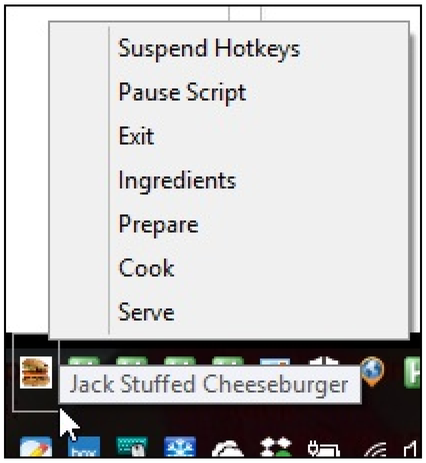
Although a little pricey, the Microsoft Surface Pro (or another less expensive Windows ultralight laptop computer) might do the job. Large enough to read the screen, yet easy to move around a cooking area, these computers help any budding chef or dietician who needs to check recipes—as long as no one dumps boiling water on the keyboard. For size and convenience, finding equipment more practical than a lightweight laptop proves difficult.

While my smartphone works great for finding specific recipes, I don't like needing to pick up the device—plus, messy fingers make poor touchscreen tools. I would prefer a standalone computer in the kitchen, but (full disclosure) I don't have one. So, this blog discusses a hypothetical situation where the cook actually maintains a Windows computer in the food preparation zone.

Note: As an alternative (and possibly the most practical approach), printing a recipe on paper eliminates any concern about destroying either the computer or a cookbook. You can easily replicate a smudged piece of paper—but you don't need AutoHotkey for that.

Jack Stuffed Cheeseburgers

I put together this barebones AutoHotkey recipe script which displays the steps for making "Jack Stuffed Cheeseburgers." Any novice AutoHotkey user can easily modify the script to work with other recipes. The app uses a cool technique which allows the cook to jump directly to any step, anywhere in the series (Ingredients, Prepare, Cook, or Serve). See the pop-up menu at the right. After selecting a step, the remaining actions automatically display after closing each successive window.

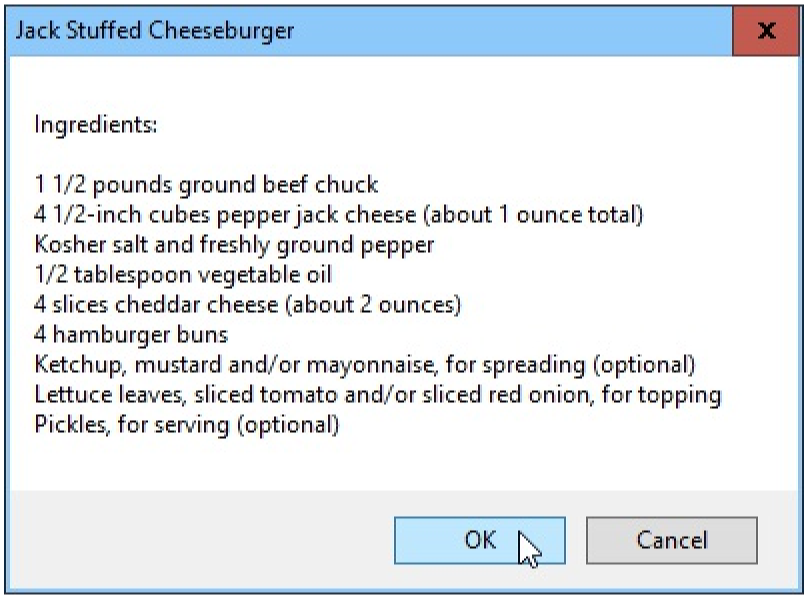


* * *

New to AutoHotkey? Check out "Installing AutoHotkey and Writing Your First Script."

* * *

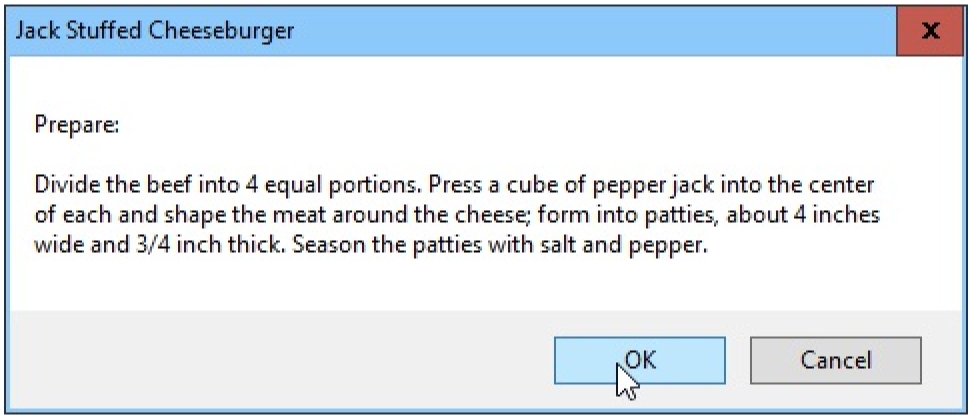
When browsing recipes with a smartphone, I found that the Web sites can get pretty annoying. Forget that the phone turns off every few seconds. The screen movement and pop-up ads often make reading anything a challenging experience. That's why I looked for a stable method for exhibiting cooking steps on a Windows computer. In this recipe script, the standard AutoHotkey MsgBox command uses the built-in Windows dialogue box to display text. After selecting the first step (Ingredients), the *MsgBox* pops up the list embedded in the script.



```
Ingredients:

1 1/2 pounds ground beef chuck
4 1/2-inch cubes pepper jack cheese (about 1 ounce total)
Kosher salt and freshly ground pepper
1/2 tablespoon vegetable oil
4 slices cheddar cheese (about 2 ounces)
4 hamburger buns
Ketchup, mustard and/or mayonnaise, for spreading (optional)
Lettuce leaves, sliced tomato and/or sliced red onion, for topping
Pickles, for serving (optional)
```

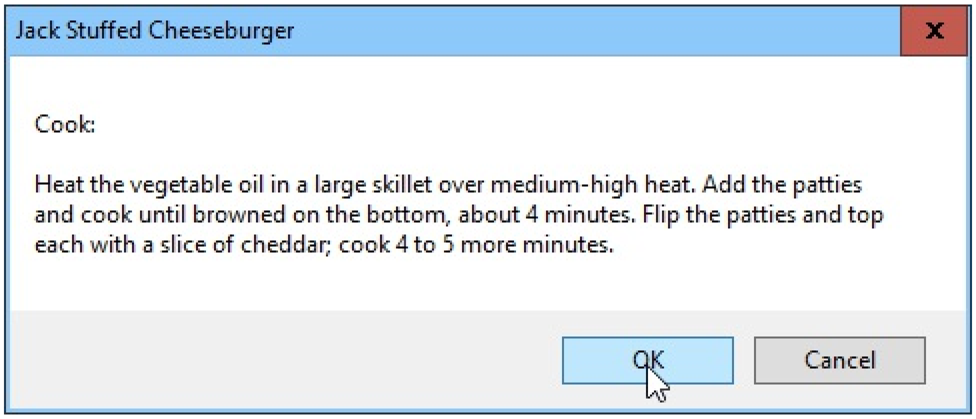
After clicking *OK*, rather than stopping, the script continues on to the next step (Prepare). This requires no special AutoHotkey tricks since each *MsgBox* occurs in sequence in the script.


Jack Stuffed Cheeseburger

Prepare:

Divide the beef into 4 equal portions. Press a cube of pepper jack into the center of each and shape the meat around the cheese; form into patties, about 4 inches wide and 3/4 inch thick. Season the patties with salt and pepper.

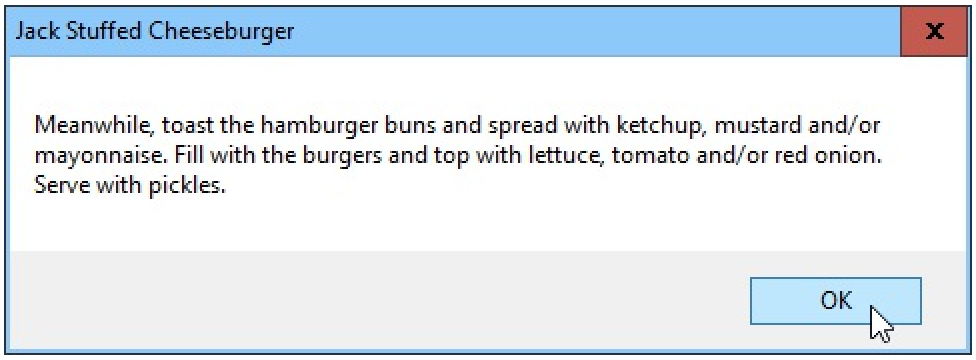
The real trick involves jumping directly to any step without passing through all the preceding *MsgBox* windows. Merely select the target item from the System Tray icon right-click menu. This works through a simple, yet little understood, characteristic of AutoHotkey Label names.


Jack Stuffed Cheeseburger

Cook:

Heat the vegetable oil in a large skillet over medium-high heat. Add the patties and cook until browned on the bottom, about 4 minutes. Flip the patties and top each with a slice of cheddar; cook 4 to 5 more minutes.

You can easily adapt this script to any other recipe with copy-and-paste in any text editor (Notepad). Replace the appropriate text between the two parentheses with that matching section copied from a Web recipe page (or any other digital source). See the script for Jack Stuffed Cheeseburger at ComputorEdge AutoHotkey script downloads.


Jack Stuffed Cheeseburger

Meanwhile, toast the hamburger buns and spread with ketchup, mustard and/or mayonnaise. Fill with the burgers and top with lettuce, tomato and/or red onion. Serve with pickles.

For those who like to get into the weeds of AutoHotkey scripting, I discuss the details of the tricks in the Cheeseburger.ahk script in my blog, Jack's AutoHotkey Blog. (Tip: The Cheeseburger recipe blogs are marked with the 🍔 image.) You can add many more features to this script through the *MsgBox* techniques discussed in "Tips for Optimizing the Standard AutoHotkey Message Box (MsgBox) Command

(AutoHotkey Quick Reference Part Six)." I started out keeping this script a little simpler, but, to demonstrate more AutoHotkey techniques, have since added printing (for those who can't afford a computer in the kitchen) and more varied recipes (Animal-Style Cheeseburger!) to the script.

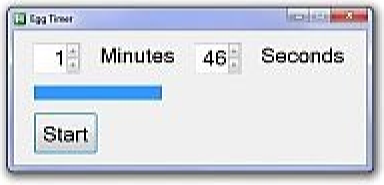
More Nutrition and Recipe AutoHotkey Scripts

In past years, I've written a number of scripts for the food category. Most of these exist for the purpose of demonstrating particular AutoHotkey commands or techniques. (However, I actually did use the calorie counting script for a number of months and lost a good bit of weight. But, alas, as with most diets, it fell into disuse—although I have not regained the lost weight.) I include the *Egg Timer* script which turns any computer into a kitchen timer. (Overkill?) The *Calorie Count* script extracts calories for specific foods from a Web page, then totals and tracks the data, then saving it to a file. Lastly, the *Recipe Tree* script demonstrates a more complex method for saving and displaying your favorite culinary innovations.

The Egg Timer

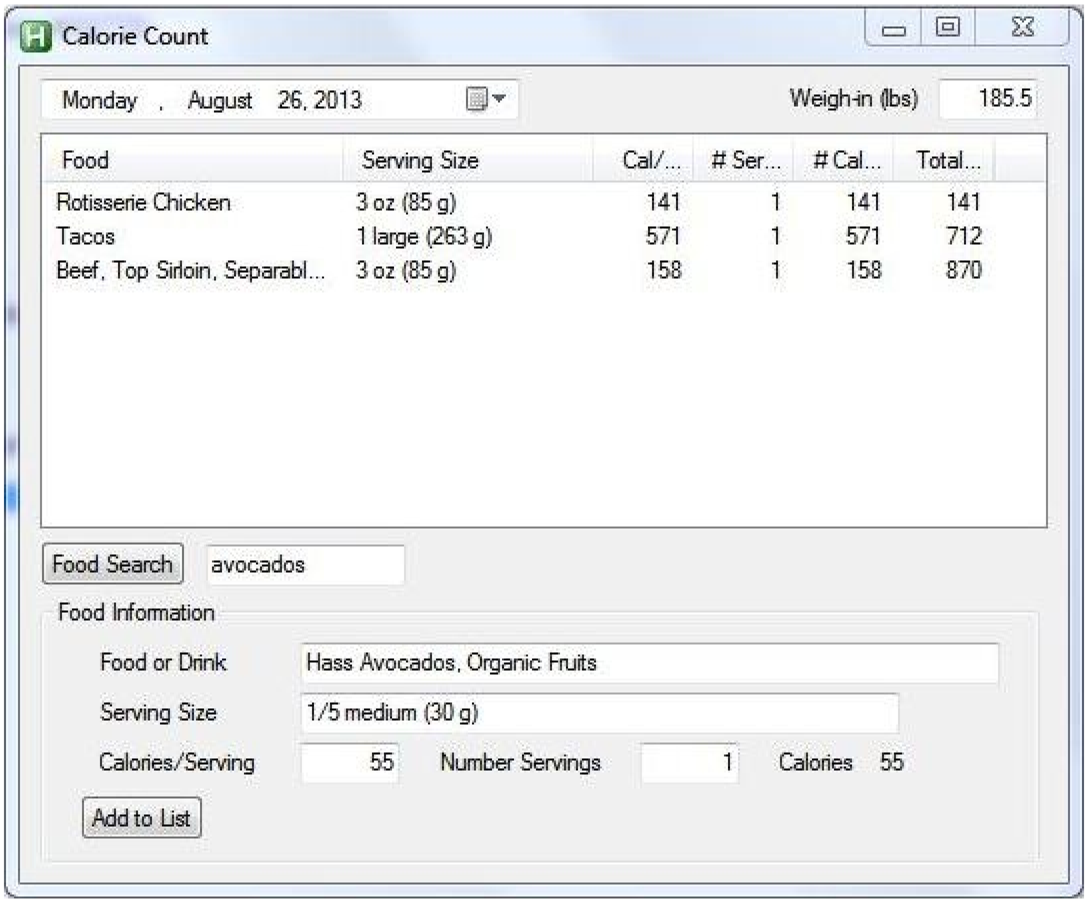
The EggTimer.ahk script displays and times any cooking process. In the *AutoHotkey Applications* e-book, I discuss the details of this script with emphasis on the SetTimer command.

The *EggTimer* app works by opening a window (after loading, press CTRL+F12) where you set the desired time interval (default three minutes), then click *Start* (see image at right). The timer counts down to zero in the edit windows (minutes and seconds) while a progress bar works it way from left to right. Once the countdown increments down to zero, the bell rings, the computer voice says "Your eggs are ready!" and a window pops up with the same message. You now have the perfect soft boiled egg.



Counting Calories

A more complex script, CalorieCount.ahk highlights the AutoHotkey ListView command— logging and tracking daily calorie counts. Find a discussion of these *CalorieCount.ahk* techniques in the *AutoHotkey Applications* e-book.



Monday , August 26, 2013 Weigh-in (lbs) 185.5

Food	Serving Size	Cal/...	# Ser...	# Cal...	Total...
Rotisserie Chicken	3 oz (85 g)	141	1	141	141
Tacos	1 large (263 g)	571	1	571	712
Beef, Top Sirloin, Separabl...	3 oz (85 g)	158	1	158	870

avocados

Food Information

Food or Drink: Hass Avocados, Organic Fruits

Serving Size: 1/5 medium (30 g)

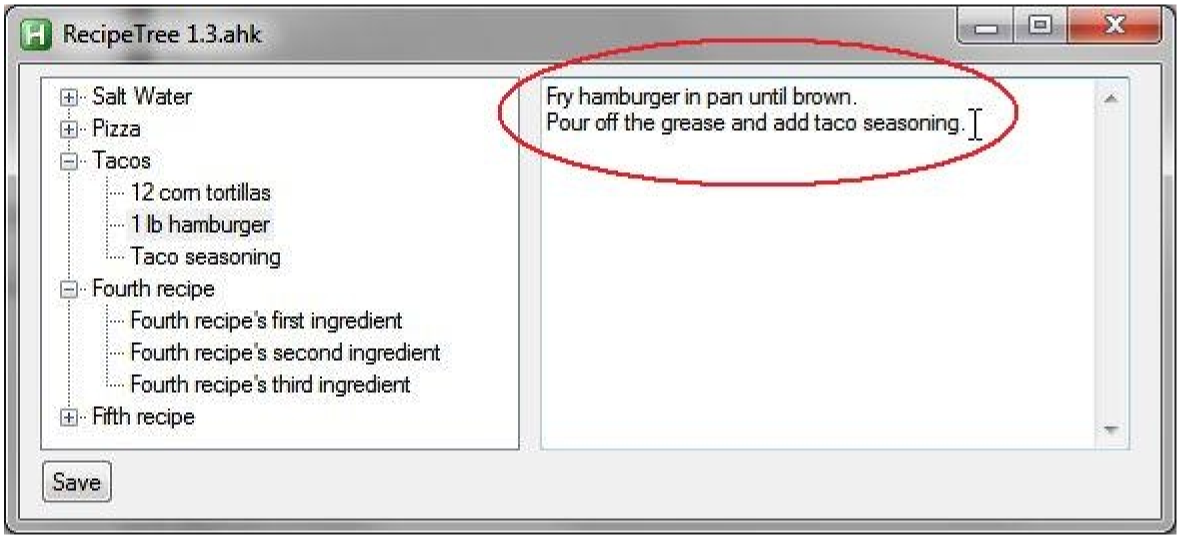
Calories/Serving: 55 Number Servings: 1 Calories 55

This AutoHotkey script keeps a daily log of calories consumed. The script demonstrates the possibilities when using AutoHotkey GUI (Graphical User Interface) windows. The GUI window embeds ListView, ListBox, GroupBox, and DateTime controls, as well as, the more standard Text, Edit, and Button controls. The script offers a special feature for importing specific food calories directly from the Internet, plus saves all data to a file.

You can convert this type of script for tracking other nutritional information such carbohydrates, fats, vitamins, and minerals.

Recipe Book

The RecipeTree.ahk script demonstrates more AutoHotkey possibilities by implementing the TreeView GUI control. Also included in the *AutoHotkey Applications* e-book, the script uses a tree format similar to Windows Explorer for tracking and displaying multiple recipes. The six-part series in the book demonstrate an alternative application for the *TreeView* AutoHotkey GUI control.



Fry hamburger in pan until brown.
Pour off the grease and add taco seasoning.

This AutoHotkey script uses a fairly advanced technique for attaching *TreeView* branches to GUI Edit control data fields by saving variable names within variables. Not a beginning AutoHotkey script, the techniques used here offer insight into building more complex applications.

Why AutoHotkey for Grandparents?

Remember All Your Grand Kid's Birthdays and Their Ages! There's No Limit to the Number of Ways You Can Amuse Your Grandchildren with AutoHotkey, Plus It Gives Your Brain a Much Needed Workout!

If you only have one grandchild, then you probably won't have much trouble recalling his or her birthday or age. In that case, you may not have much interest in the little AutoHotkey GrandKids.ahk script. However, AutoHotkey offers much more which can enrich your offspring's offspring's education and entertainment—including a one-line script which verbalizes out loud the letters and numbers on the computer keyboard. But more importantly, learning to write AutoHotkey scripts exercises your mind—something everyone needs.

"I'll be glad when we upgrade our Grand Bot's operating systems beyond Terrible Two Point Oh!"

* * *

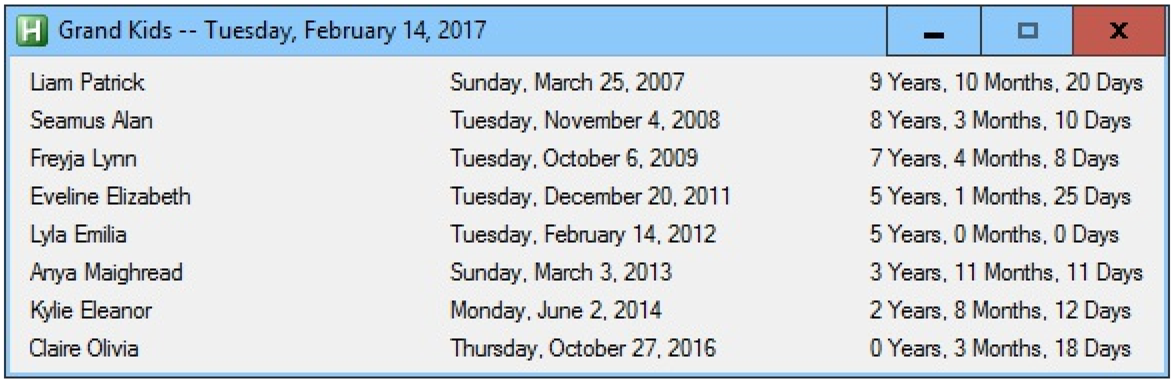
Some of the scripts in this blog may not make AutoHotkey look easy, but you'll find the first steps to AutoHotkey literacy quite simple. For a comfortable startup, check out this Introduction to AutoHotkey.

* * *

One of the most daunting tasks for any grandparent involves remembering and acknowledging the birthdays of your grandchildren—especially if you have a plethora of them. Forgetting the birth of a grandkid falls among the greatest sins that a family elder can commit. You may need a computer to keep track of them all. Fortunately, AutoHotkey makes it easy.

Remembering Birthdays and Ages

You can use the calendar on a computer or smart phone to remind you of approaching birthdays, but they don't usually tell you the age of the honoree. Even then, to find the exact day, you must sift through the months searching for each birthday. What if you could add all the names and dates to one short file, then run an app which listed all of the birthdays and the exact age of each child (down to the day)? The AutoHotkey GrandKids.ahk script does that!



Grand Kids -- Tuesday, February 14, 2017

Liam Patrick	Sunday, March 25, 2007	9 Years, 10 Months, 20 Days
Seamus Alan	Tuesday, November 4, 2008	8 Years, 3 Months, 10 Days
Freyja Lynn	Tuesday, October 6, 2009	7 Years, 4 Months, 8 Days
Eveline Elizabeth	Tuesday, December 20, 2011	5 Years, 1 Months, 25 Days
Lyla Emilia	Tuesday, February 14, 2012	5 Years, 0 Months, 0 Days
Anya Maighread	Sunday, March 3, 2013	3 Years, 11 Months, 11 Days
Kylie Eleanor	Monday, June 2, 2014	2 Years, 8 Months, 12 Days
Claire Olivia	Thursday, October 27, 2016	0 Years, 3 Months, 18 Days

The GrandKids.ahk app lists birthdays and the age of each child calculated to the day. (This script works equally well for any other commemorative dates.)

Activate the Hotkey combination and a window pops up—automatically calculating the ages.

As you can see, Lyla turned five on Valentine's day in 2017 (*5 Years, 0 Months, 0 Days*). (The surprise came when I noticed the fast approach of Anya's birthday.) This app comes in handy when you ask "Now, how old is he/she?" AutoHotkey uses the names and dates stored in an INI file (**GrandKids.ini**) which opens with any text editor (Notepad?).

I take particular pride in the AutoHotkey function for calculating the difference between two dates in years, months, and days (*HowOld(FromDay,ToDay)*). Working with dates always presents a challenge since the increments (e.g. number of days or months in a year, number of days in each month) are not particularly rational. I struggled with the date calculations until I finally nailed the function. (Not for the faint of heart, but the mental exercise added one year to my brain life. Three chapters included as one section of the e-book *Digging Deeper Into AutoHotkey* explain the evolution of the *GrandKids.ahk* script and the *HowOld()* function.)

For a simpler AutoHotkey app, let's make the keyboard talk.

A Simple One-Line Script to Teach the Computer Keyboard and Alphabet

Kids love noise—even if educational! I've played with a number of AutoHotkey scripts which activate the Windows voice feature to both read sentences and repeat keyboard key names (more on that later). But, only the other day I came up with an elegant one-line solution for turning the Windows keyboard into an educational tool. (Actually, you might need to add one additional line to a stand-alone script since AutoHotkey must install the keyboard hook for the A_PriorKey variable to work.)

When running an AutoHotkey script with the keyboard hook installed, the last key pressed automatically gets stored in the variable *A_PriorKey*. (I didn't realize this until a recent exploration of the online AutoHotkey documentation.) In the past, I had worked on keyboard scripts which required me to list each individual key with a new line of code. (Tedious.) Now, with the *A_PriorKey* variable, I can condense it all down to a one-line Hotkey:

```
~Space::ComObjCreate("SAPI.SpVoice").Speak(A_PriorKey)
```

Once loaded, a grandchild can push any keyboard key, then press the SPACEBAR. The computer speaks the key out loud. While this seems incredibly mundane, for grandchildren getting their hands on Grand Dad's "big" computer is a huge deal.

(Okay, maybe understanding the command isn't all that simple, but it is only one line.)

The squiggle (~) before the SPACEBAR key (~*Space*) causes the key's purpose to pass through to its original function—entering a space into any document—without getting blocked. That way you can continue writing and editing without exiting the script. However, every time you hit the SPACEBAR, the computer speaks the last character you keyed. To prevent this, you may want to use another less important key such as CAPSLOCK:

```
~CapsLock::ComObjCreate("SAPI.SpVoice").Speak(A_PriorKey)
```

Note: While you don't need to understand exactly how the keyboard hook works, it must install before using the *A_PriorKey* variable. In scripts, the appearance of any Hotstrings or Hotkeys automatically installs the keyboard hook or you can set it unconditionally with the follow directive:

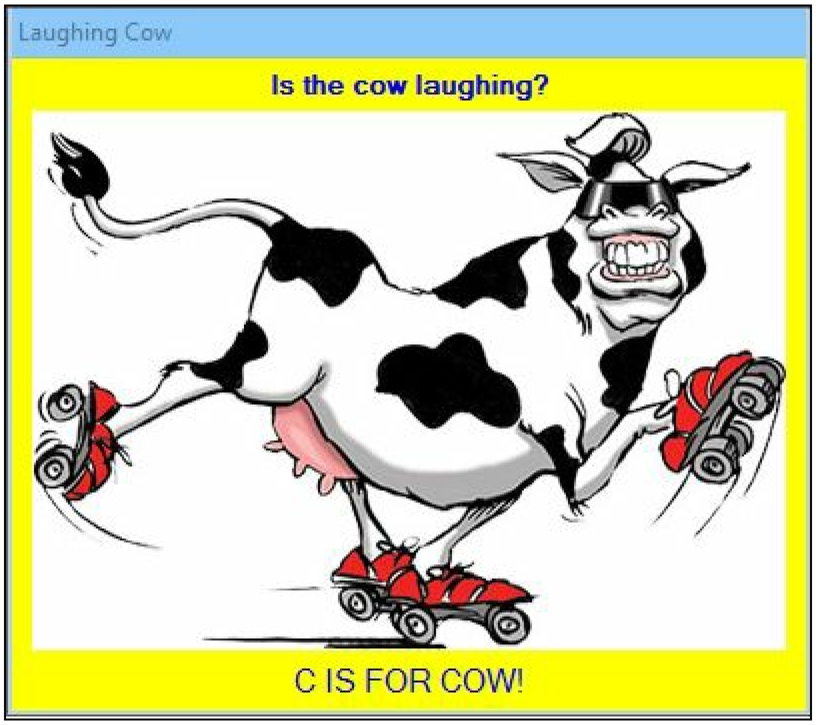
```
#InstallKeybdHook
```

Fortunately, since our little one-line script creates a Hotkey (~*Space::* or ~*CapsLock::*), the keyboard hook loads automatically without needing the *#InstallKeybdHook* directive.

(While AutoHotkey does a great deal more, Hotstrings and Hotkeys rank as the two most important reasons beginners take up AutoHotkey. I've published two recent e-books devoted to the topics: *Beginning AutoHotkey Hotstrings* and *AutoHotkey Hotkey Techniques*.)

Once you get into AutoHotkey (which you should for the sake of your mental longevity), you'll find even more ways to entertain and educate the youngsters.

More AutoHotkey Entertainment and Education



The SayWhat.ahk script demonstrates how you can easily use the text reading capability of Windows in an AutoHotkey script. Once loaded, the script opens the "Speak to Me!" window. Type in any text, then click "Talk!" The default Windows voice reads the text out loud.

NumbersSpeak.ahk contains, *cow-skating.jpg* (image file) and *cow-madcow.wav* (audio file). The *NumbersSpeak.ahk* script uses the *SayWhat.ahk* script as a basis for exploring techniques in writing children's educational software. Press the letter *C*, and the script displays the *SplashImage* (shown above), reads the words "C is for cow!", then spells “c – o – w”, followed by the sound of a laughing mad cow mooing. Using the SplashImage and SoundPlay commands, plus the ComObjectCreate() function *ComObjCreate(“SAPI.SpVoice”).Speak()*, this example had my grandkids rolling in the aisles. (I discuss this script in Chapter Thirteen of *AutoHotkey Hotkey Techniques*.)

While far from complete, the script features some of the AutoHotkey tools available for building cute children’s scripts. It's an apt companion for the educational TalkingText.ahk script which voices the letters of the keyboard, then makes animal sounds when spelling certain animal names (e.g. cat, dog, bear, and more).

Exercise the Brain

While there have been a number of studies about what happens to the human brain in later years, most are inconclusive—especially when it comes to what to do about it. It seems that regular physical exercise helps, but the jury is out on how much various types of mental stimulation assist. However, most articles conclude that any form of cerebral gymnastics is “better than nothing” or “It can’t hurt!”

From Time, “Games sure seem like a good way to work your brain out, but don’t put your stock in Sudoku. ‘They target very specific cognitive abilities, but they don’t transfer to clarity of thinking, problem solving, planning—all the complex skills that really matter,’ explains Dr. Sandra Bond Chapman, chief director of the Center for Brain Health at the University of Texas at Dallas and author of *Make Your Brain Smarter*.”

“Along with aerobic exercise, engaging your brain in complex ways is absolutely necessary to keep your mind sharp in the second half of life.” Studies have found few links between doing puzzles such as crossword and Sudoku and staving off mental deterioration. However, it may be that learning to write programs or computer scripts offers everybody increased cerebral longevity.

From the article, “Can Learning to Code Delay Alzheimer’s?“”

> "Several studies have shown that being bilingual or multilingual significantly delays the manifestation of Alzheimer’s because bilingual brains have greater adaptability and functionality. So are programmers’ brains similarly resistant to developing the disease?"

Professor Janet Siegmund of the University of Passau and her colleagues ran fMRI brain scans on 17 volunteers while they were reading code snippets for a study in 2014.

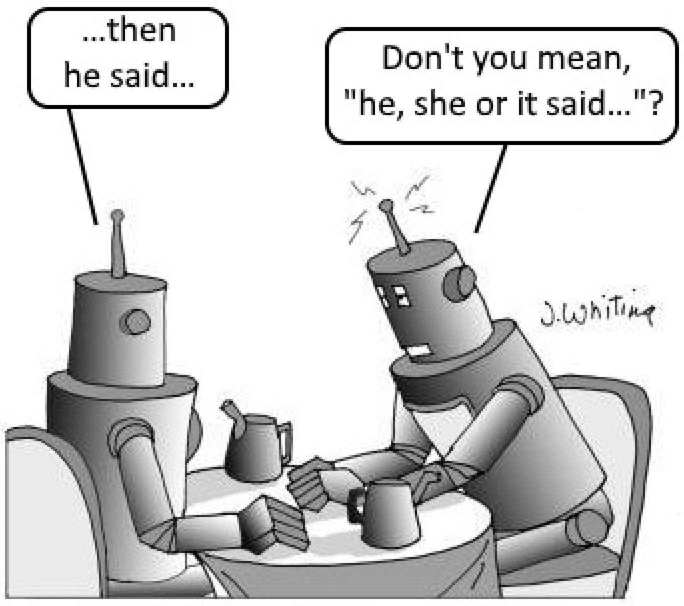
“We found [the] first empirical evidence that both, natural language and programming language, require the same areas in the brain,” she said. “Based on this, we can infer that understanding programming languages and natural languages appear to be similar.”

While not definitive, I’ve decided *not* to wait until science proves whether or not mental stimulation (and which type) actually helps keep my brain in shape. I’ll continue doing crossword puzzles (and cheating via the Web) and exploring new ways to apply AutoHotkey on my Windows computer. I recommend that you do the same.

Why AutoHotkey for Students?

Ten Reasons Why Every Student Should Use the Free AutoHotkey Windows Tools! AutoHotkey Helps You So Much Academically That You'll Think You're Cheating! Plus, It's Delicious Brain Food and Helps Get You a Job!

Using AutoHotkey should be mandatory for every student from high school on through college! Not merely because learning AutoHotkey invigorates the brain cells (which it does!), but it makes all your Windows tasks so much easier that it seems unsporting. AutoHotkey takes the grunt work out academia making learning fun…almost.



Robot Sexism

(If you know a student who uses a Windows computer, then do him or her a favor by telling him or her about AutoHotkey. If he or she doesn't already know about it, he or she will thank you profusely.)

For those unfamiliar with the free open-source scripting language, AutoHotkey offers simple, yet powerful, commands for adding enhancements to any Windows computer. In addition to their simplicity, the little apps work in any Windows program or on any Web page. For a brief overview, see this "Introduction to AutoHotkey: A Review and Guide for Beginners."

Ten Reasons Students Should Use AutoHotkey

The AutoHotkey Windows scripting language assists scholars in far more ways than the ten listed here, but this represents a good start.

1. AutoCorrect on the Fly Most Commonly Misspelled English Language Words

2. Write Gender Neutral Essays and Term Papers

3. Insert Funky Foreign Language Characters in Your Papers

4. Instantly Add the Date in Any Format to Any Document

5. Quickly Collect Quotes and References for Research Papers

6. Protect Yourself from Lost Work with Personal Backup

7. Use Instant Hotkeys to Save Typing Time While Adding Complex Phrases and Expressions

8. Eliminate Student's Most Commonly Overused Words in Papers

9. Quickly Find Even More Synonyms

10. Improve Your Mind and Get a Job

I don't expect people who are new to AutoHotkey to immediately understand the inner workings of all these techniques—although, the first tricks are incredibly simple. In fact, making AutoHotkey part of your academic life is a journey. Just start at the beginning and take one step at a time. With the simple noobie techniques, you'll get immediate results. Soon, any investment of time pays great dividends.

Reason One: AutoCorrect on the Fly the Most Commonly Misspelled English Language Words

Hotkeys and Hotstrings put the "Hot" in AutoHotkey. Each technique uses double colons to designate either an action to execute (Hotkey) or an instant text replacement (Hotstring). A Hotkey use one double colon (*[Hotkey combination]::[AutoHotkey command]*):

```
^!O::Run, http://www.autohotkey.com/
```

In this one-line example, when added to an AutoHotkey script and loaded with the AutoHotkey program running, the Hotkey combination CTRL (^) + ALT (!) + O (pressed simultaneously) loads the *http://www.authotkey.com* Web page in the default browser on any Windows computer.

For Hotstring text replacement, we use two sets of double colons (*::[Hotstring]::[Replacement Text]*):

```
::lol::laugh out loud
```

When loaded, type "lol" followed by punctuation to instantly get "laugh out loud" in any Windows document or text field. The AutoHotkey *AutoCorrect* app consists of thousands of these independent Hotstring lines.

I often recommend the AutoCorrect.ahk script before any other AutoHotkey app. My slightly modified version of the *AutoCorrect.ahk* script originally came from the AutoHotkey Download Web site. While typing in any Windows program or Web editing page, it instantly corrects commonly misspelled English words. See "Add AutoCorrection to All Your Windows PC Programs with AutoHotkey."

AutoCorrect works through the built-in AutoHotkey Hotstring feature. Following a misspelled word with either punctuation or a space immediately replaces the errant term with the correct word. For example, the code line:

```
::mispelling::misspelling
```

yields *mispelling→misspelling*

The code line:

```
::europians::Europeans
```

yields *europians→Europeans*

At the Free ComputorEdge AutoHotkey Apps page, the file download *AutoCorrect.zip* contains *AutoCorrect.ahk* and the compiled version (which runs on any Windows computer without AutoHotkey installed) *AutoCorrect.exe*. (You can find a discussion of the *AutoCorrect.ahk* script in the e-book *Digging Deeper Into AutoHotkey*.)

Note: The AutoHotkey AutoCorrect script is neither spelling nor grammar correction software. In fact, the complications of the grammar correction in English create unique scripting issues. Fortunately, plenty of other programs, such as the free version of Grammarly grammar correction software, do just that.

Reason Two: Write Gender Neutral Essays and Term Papers

With the current state of political correctness on college campuses, it behooves any student—regardless of political leanings—to keep their class submissions as gender neutral as possible. The following short starter script, similar to *AutoCorrect.ahk*, changes any sexist language while you type into more acceptable terminology. Depending upon the course and instructor, this may turn out to be your most important use of AutoHotkey.

These Hotstrings are by no means a complete list but they represent a good start:

```
::man*::man ; use * if you really mean man and not person
::men*::men ; use * if you really mean men and not people
:b0:woman:: ; prevents woman from becoming woperson
:b0:women:: ; prevents women from becoming wopeople
::firemen::fire department
::policemen::police
::mailmen::postal workers
::mailman::postal worker
::postmen::postal workers
::postman::postal worker
:?:man::person
:?:men::people
::he::he or she
::him::him or her
::his::his or her
:c:Miss::Ms
:c:Mrs::Ms
::mankind::humankind
```

The priority of a Hotstring depends upon its placement in the file. The first appearance of a similar Hotstring overrides any following Hotstrings. For example, with the script above loaded, when typing *woman*, the statement *:b0:woman::* with no backspace (option *B0*) activates first and prevents the *:?:man::person* Hotstring from firing and yielding the replacement *woman→woperson*. In another example, "firemen" gets replaced with "fire department" rather than "firepeople".

Using the question mark *?* option between the first set of double colons activates replacements at the end of words. The *c* option forces matching of capital letters in the activating text. However, the question "Miss me?" produces "Ms me?" (You'll need to fix that one by hand.) Clicking the left-mouse button or hitting the ESC key prior to activation of the Hotstring cancels its action.

This list of Hotstrings merely offers examples of how you can implement instant gender neutral AutoCorrection into your school work with AutoHotkey. Please feel free to modify this script as necessary to conform to your professor's standards. He or she will appreciate the effort—even though AutoHotkey makes it effortless.

Reason Three: Insert Funky Foreign Language Characters

This little AutoHotkey Accent app from Skrommel comes in handy if you work with foreign languages or when you want special characters for the few English words which ask for them (e.g. résumé). Since these special characters don't appear on most keyboards *Accents.ahk* makes is much easier to add various types of squiggles to your documents.

The script makes clever use of the AutoHotkey Input command and an INI data file to make inserting the special characters simple. For example, press the letter "a" key three times to start cycling through the list (*å*, *ä*, *â*, and *à*) with each press of the key. While this script currently only works with the vowels, with very little effort other letters can be added.

I found that including different characters (e.g. n→ñ) simply requires adding them to the INI file. For example, I added the Spanish ñ by inserting the following as the seventh key:

```
[7]
key=n
1=ñ
2=n
```

Bug Report: When running, the *Accent* app interfered with my loaded Hotstrings. (Go figure.) I believe that the *Input* command takes over the monitoring of the keyboard. Although the script offers an ingenious use of the Input command, since in its current form *Accents.ahk* disables other Hotstrings, I've added a *Suspend* option to the script in the System Tray icon right-click menu. I recommend suspending the script whenever you don't need the alternative accents. In the long run (and for anyone who wants to take on the task), a Hotkey to temporarily activate the routine or a pop-up menu might work better without blocking other active Hotstrings. I'd recommend only loading the *Accents.ahk* script when you need it.

While the script needs a little work, it's a good idea which could easily be expanded to other applications.

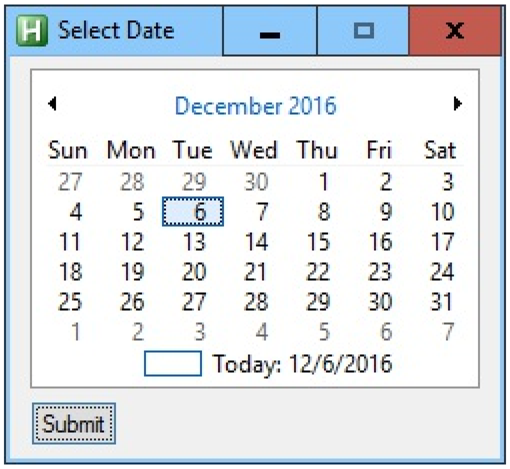
Reason Four: Instantly Add the Date to Any Document

"Don't forget to date your papers!"

One Hotstring I use regularly inserts today's date into any document. I simply type *Anow* (capital *A* required). The Hotstring instantly pastes the current date (e.g. January 27, 2017) into any Windows edit field. AutoHotkey only requires four lines of code:

```
:c*:Anow::
  FormatTime, CurrentDateTime,, MMMM d, yyyy
  SendInput %CurrentDateTime%
Return
```

The ComputorEdge Free AutoHotkey Scripts page offers various techniques for inserting dates—including a pop-up calendar. We updated the MonthCal GUI (Graphical User Interface) script, *AddDate.ahk* (a demonstration script for adding dates to the current document), to deal with a common AutoHotkey active window error. You'll find discussions of the Hotkeys included in the book, A Beginner's Guide to AutoHotkey, plus the two Hotstrings in the AutoHotkey Applications e-book.)



When loaded, the script demonstrates four different methods for adding the date with Hotkey combinations:

- CTRL+WIN+F1 ⇒ The datetime stamp (e.g. 20130628103349)
- CTRL+WIN+F2 ⇒ The date using the *FormatTime* command (*FormatTime, TimeString, %A_NOW%, MMMM d,* yyyy)
- CTRL+WIN+F3 ⇒ The date using built-in variables (*%A_DDD%, %A_MMM% %A_D%, %A_YYYY%*)
- CTRL+WIN+D ⇒ The *MonthCal* GUI pop-up calendar (*Gui, Add, MonthCal, vDayPick*)

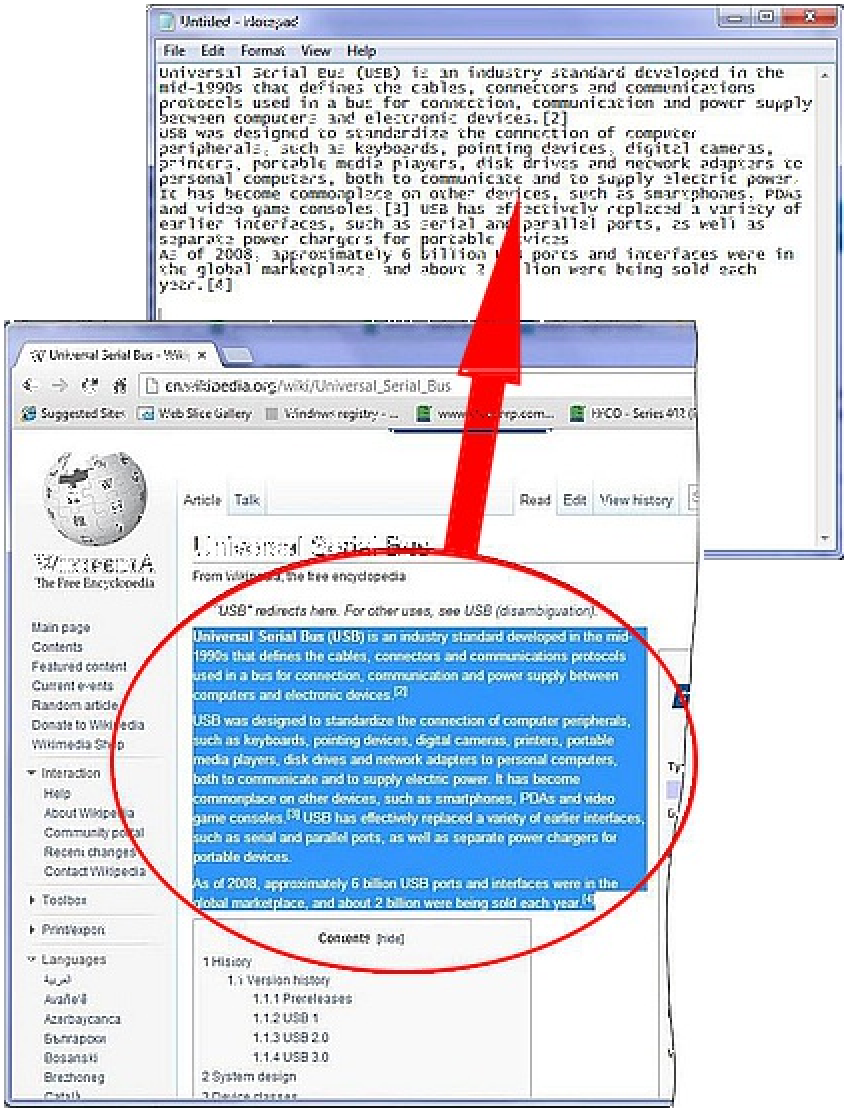
plus two Hotstrings:

- *Anow* ⇒ Instantly add today's date formatted (e.g. September 7, 2013)
- *Adate* ⇒ Activate *MonthCal GUI* pop-up

Reason Five: Collect References for Research Papers

The first step in writing any research paper requires digging into references. While the Internet should not act as your only source, it makes a good starting point. Web searches help to identify and validate possible sources—as long as you don't get caught up in the tautology of parroted Web pages. The most tedious research tasks involve extracting and compiling quotes—along with the reference page addresses. I put together a quick AutoHotkey tool which allows you to copy text and reference data from any source send it to a Notepad window without ever leaving your research work.

This CopyRef.ahk script acts as an AutoHotkey reference collection tool for any research work. Highlight text on any page or in any document and copy to a Notepad collection window with the Hotkey combination CTRL+ALT+C. If not already open, the script launches and displays an "Untitled" Notepad window. Then, whenever executing the Hotkey combination, the app continues to collect the next selected text in the same Notepad window without activating it—even if minimized. This saves the repetitive copy-and-paste routine while switching between windows.



```
Universal Serial Bus (USB) is an industry standard developed in the
mid-1990s that defines the cables, connectors and communications
protocols used in a bus for connection, communication and power supply
between computers and electronic devices.[2]
USB was designed to standardize the connection of computer
peripherals, such as keyboards, pointing devices, digital cameras,
printers, portable media players, disk drives and network adapters to
personal computers, both to communicate and to supply electric power.
It has become commonplace on other devices, such as smartphones, PDAs
and video game consoles.[3] USB has effectively replaced a variety of
earlier interfaces, such as serial and parallel ports, as well as
separate power chargers for portable devices.
As of 2008, approximately 6 billion USB ports and interfaces were in
the global marketplace, and about 2 billion were being sold each
year.[4]
```

Universal Serial Bus (USB) is an industry standard developed in the mid-1990s that defines the cables, connectors and communications protocols used in a bus for connection, communication and power supply between computers and electronic devices.[2]

USB was designed to standardize the connection of computer peripherals, such as keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters to personal computers, both to communicate and to supply electric power. It has become commonplace on other devices, such as smartphones, PDAs and video game consoles.[3] USB has effectively replaced a variety of earlier interfaces, such as serial and parallel ports, as well as separate power chargers for portable devices.

As of 2008, approximately 6 billion USB ports and interfaces were in the global marketplace, and about 2 billion were being sold each year.[4]

The Hotkey combination CTRL+ALT+C immediately pastes the highlighted (selected) text into the Notepad window.

The script inserts the text at the last cursor location in the target window. New lines are added after each copy action. This reference collecting script works with any Windows source which renders text, whether Web pages, word processing documents, e-books in any form (e.g. EPUB, PDF, MOBI), or e-mails.

I wrote this script with students in mind, but, of course, any profession can use it. The *CopyRef.ahk* script represents a much more robust version of the original SaveText.ahk script used to copy selected text to the AutoHotkey app ScratchPad by Desi Quintans (discussed in Chapter Seventeen of the e-book *Digging Deeper into AutoHotkey*). In addition to a number of additional protections, the *CopyReg.ahk* script uses the standard AutoHotkey Clipboard manipulation routine, plus, adds the Control, EditPaste command for greater speed and reliability.

Reason Six: Protect Yourself from Lost Work

Have you ever been working on a clever post or comment when the window suddenly closes, the computer freezes, or you accidentally delete your work? All your brilliant efforts disappear. There's nothing more frustrating—especially if you've completed a significant amount of work.

Blogging sites such as WordPress include regular automatic backup, but most of the time (Facebook and other sites) you're on your own. Now with AutoHotkey, you can instantly save your work at any time with your own temporary or incremental backup. If something goes wrong or your computer freezes, you recover data with a backup file.

At "Free AutoHotkey Scripts and Apps for Learning Script Writing and Generating Ideas" you'll find two sample instant backup scripts: BackupText.ahk and IncrementalSaveText.ahk.

The *BackupText.ahk* script copies selected text to the file *SaveEdit.txt* located in the user's *Documents* folder. With each use of the CTRL+ALT+B Hotkey combination, AutoHotkey selects all the text (CTRL+A) in the document or Web editing field, copies it to the Windows Clipboard, then saves it to the *SaveEdit.txt* file. Only use this script for quick, temporary backup of current unsaved work. The script overwrites the old file on each Hotkey activation. For incremental backup to text files use the *IncrementalSaveText.ahk* script which follows.

The *IncrementalSaveText.ahk* script discussed in Chapter Fifteen of the *Digging Deeper Into AutoHotkey* saves the text from any window, including Web pages, by selecting the text, copying it to the Windows Clipboard, then saving it in a backup text file with a filename which includes the window's title plus the current date and time in date/time format (e.g. *Hotstrings and Auto-replace (similar to AutoText and AutoCorrect) – Google Chrome20170129121029.txt* for the AutoHotkey Hotstrings page). The script saves the file in the user's *Documents/Backup* folder. If the *Backup* folder doesn't exist, AutoHotkey creates it. Once loaded, CTRL+WIN+ALT+B initiates the script's backup procedure.

(If you want the backup to occur automatically, then call the routine with the SetTimer command.)

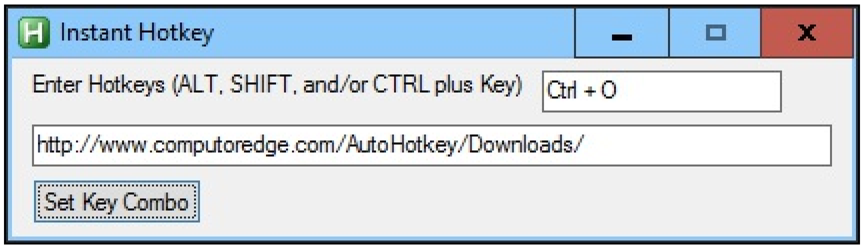
Even if you use an app such as WordPress which automatically backs up your work, you still have a reason to create your own personal backup system. For example, if you accidentally delete a few paragraphs and the auto-save suddenly kicks in, your work may be lost forever. Even the Windows emergency CTRL+Z key combination may not recover it. In those situations, you'll be thankful for AutoHotkey backup.

Reason Seven: Save Typing Time with Instant Hotkeys

When working on papers certain esoteric topics may call for repeating a tedious word or phrase a number of times. However, you don't want to permanently add it to your *AutoCorrect* list. In those situations, you need a script which instantly creates an AutoHotkey Hotkey for inserting the repetitious expression.

The AutoHotkey script InstantHotkey.ahk (discussed in the AutoHotkey Applications e-book) temporarily sets up Hotkey combinations for adding repeated and/or long lines of text to any current Windows document or Web editing field.

Use this handy *InstantHotkey.ahk* script when you need to insert a particular term or set of terms into a paper numerous times. After loading the script, the Hotkey setup window immediately opens. Enter a new Hotkey combination along with the insertion text. After setting the key combination, you can change it by right-clicking on the icon in the System Tray and selecting *Show Hotkey*. Hovering over that same icon displays the current Hotkey setting. Use the key combination CTRL+ALT+H to set up as many temporary *Instant Hotkeys* as you like.



Ctrl + O

http://www.computoredge.com/AutoHotkey/Downloads/

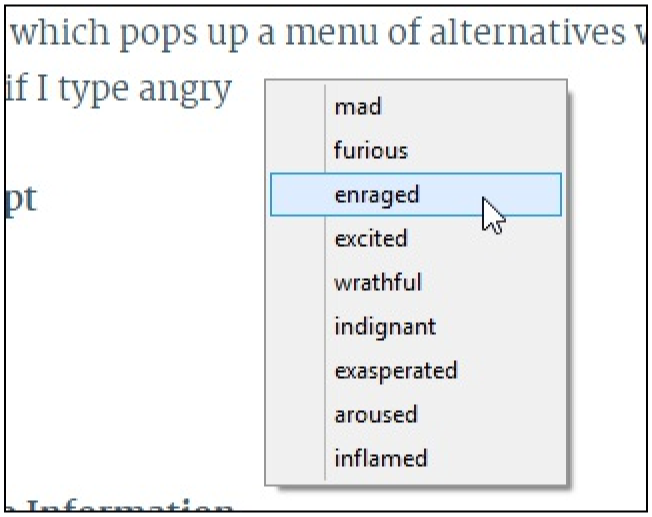
After setting up an *Instant Hotkey*, every time you need to add the text to a document, simply press the assigned keys simultaneously. For example, in the image above, the pressing of CTRL plus O at the same time inserts the URL:

```
http://www.computoredge.com/AutoHotkey/Downloads
```

Reason Eight: Eliminate Student's Most Commonly Overused Words in Papers

As a pupil, using the same, redundant words over and over again ranks at the top of term paper no-nos. After discovering a list of synonyms for those excessively abused words, I put together a pop-up menu for identifying and replacing them.

The script OverusedWords.ahk monitors the keyboard action, then popping up a menu of replacements for the list of overused words—as determined by English teachers when grading papers. If you regularly use the offending words, this script will drive you crazy as it forces you to pick alternatives. While *OverusedWords.ahk* primarily targets students, anyone who wants to improve their writing can benefit. (Discussed in Chapter Eleven of Beginning AutoHotkey Hotstrings.)



After typing "angry" in any document or editing field, a pop-up menu of alternative synonyms appears.

The *OverusedWords.ahk* script only contains problem words. If you want synonyms for any term, the following AutoHotkey technique can make Web lookups easier and quicker.

Reason Nine: Quickly Find Even More Synonyms

This short beginning AutoHotkey strip (SynonymLookup.ahk) demonstrates how to use the Windows Clipboard to quickly access reference sites on the Web. In this case, after hitting the Hotkey combination CTRL+ALT+L, *Thesaurus.com* opens in the default browser using the previously highlighted word as a key for synonyms. The SynonymLookup.ahk script demonstrates a common Web search technique which can be used with any Web page which allows searches from the URL field.

Reason Ten: Improve Your Mind and Get a Job

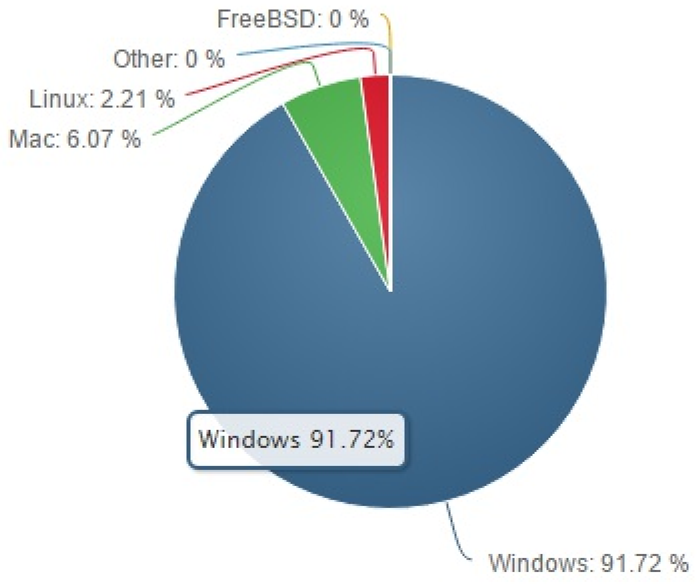
AutoHotkey as Brain Food

Sharper thinking may be the most important benefit from learning to write AutoHotkey scripts. In a world cluttered with insanity and arbitrariness, developing an understanding of how to write AutoHotkey scripts adds logic and discipline to a sloppy mind. Empowering your computer with AutoHotkey scripts changes the way your think. While you may not learn the answer to "Life, the Universe, and Everything", you will like the way AutoHotkey improves your mental acuity and eases your computing life. Plus, it's fun!

Many people think that programming is only for programmers. Not true! The beauty of AutoHotkey is that virtually anyone can learn to write simple one-line Windows enhancing Hotstrings and Hotkeys. If that were the only benefit a student received by using AutoHotkey, it would be plenty. But, as users learn more about AutoHotkey, bit by bit they venture into increasingly powerful tools. These new methods provide extra gains. In fact, even though you may start with simple scripts, the AutoHotkey language offers the depth of potential found in most programming languages—capable of producing full-featured desktop applications. Fortunately, the structure and implementation of AutoHotkey offers an ideal path for learning a little at a time.

Significantly, writing AutoHotkey scripts trains the mind in the analytical methods required in almost every phase of life. It teaches problems solving skills through the implementation of step-by-step logic paths and debugging techniques. The paramount importance of proper evaluation of the *If-Then-Else* decision model makes the analysis of any problem easier. The techniques for looping through series of alternatives gives users an understanding of the need to test all possibilities. Rather than coming out of school with a mishmash of contradictory thought processes, your mind graduates as a razor-sharp tool.

New Job Credentials



Best of all, if you learn AutoHotkey (regardless of your major), you'll graduate from school with a skill useful in almost any working environment! As shown in the graphic at the right (Source: Net Market Share), over 90% of the world's people, businesses, and organizations currently use Windows desktop and laptop computers (and that number shows no sign of decreasing). Yet, most of those computer users are totally unaware of the existence of the free AutoHotkey software. Since these scripts run on virtually all of those Windows computers, the opportunity to add value to a company or organization with profession specific tools offers unlimited opportunity for the AutoHotkey initiated.

If you can use AutoHotkey to make magic with Windows, then your value increases dramatically—much more than people who still wonder how a computer works. You don't need to be an engineer or programming student to add computer credentials to your résumé. Just start using it. Learning AutoHotkey

—even on the simplest level—gives you an edge over any others competing for the same job.

Why AutoHotkey for Teachers and Educators?

If You and Your Students Have Access to Windows Computers, Then You're Set! If Not, Well…

The varied computer situations of educators present a particular problem for AutoHotkey use. AutoHotkey works only on Windows computers. Fortunately, 90% of the desktop and laptop computers currently in use run Microsoft Windows. On the downside for teachers, they may or may not have access to Windows machines in the classroom. Plus, even if students have a computer available at home, not all of them are Windows PC. This makes it difficult to create consistent computer-dependent lesson plans covering everyone in the class.

"Look! Robbie has invented a new word processor that doesn't require a computer or batteries!"

Noting the difficulties with applying a Windows-based program to schooling, I forge on with the possibilities for those instructors in a position to take advantage of AutoHotkey.

AutoCorrect That Teaches

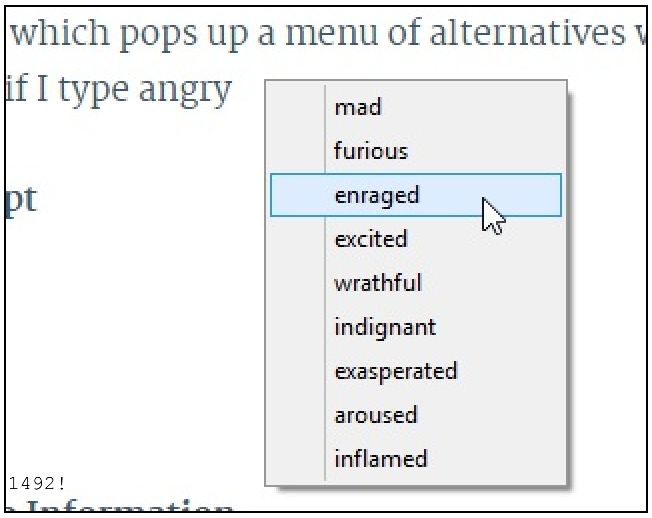
One of the most common ways to use AutoHotkey involves the implementation of the AutoCorrect.ahk script which fixes misspelled words on the fly. The slightly modified *AutoCorrect.ahk* script which came from the AutoHotkey Download Web site instantly corrects commonly misspelled English words while typing in any Windows word processor or Web editing page. See "Add AutoCorrection to All Your Windows PC Programs with AutoHotkey."

The innovative educator can change up the AutoCorrect script to include those annoying words students regularly interject into something more appropriate and/or educational. For example, suppose a teacher notices that students overuse the word "fat" in their papers. Add an alternative Hotstring to a special AutoCorrect script which might include an automatic replacement:

```
::fat::stout
```

Of course, numerous other options exist (i.e. corpulent, fleshy, beefy, paunchy, plump, full, rotund, tubby, pudgy, chubby, chunky, burly, bulky, elephantine). Other possible AutoHotkey approaches similar to the OverusedWords.ahk script might work better. This Hotstring app pops up a menu of replacements for a list of overused words—as determined by English teachers when grading papers. (See image at right which offers substitutes for the work *angry*.) While primarily for students, this script works for anyone hoping to improve their writing. Chapter Eleven of *Beginning AutoHotkey Hotstrings* discusses this script.

Another creative AutoHotkey script might use these same techniques found in the *OverusedWords.ahk* script to pop-up educational windows (MsgBox command rather than a menu of replacement words) whenever a student types specific words on their windows computer. For example, suppose you want to reinforce the idea that Columbus set sail for the new world in 1492? Include the following code in any AutoHotkey script:



```
:b0:1492::
  MsgBox, Columbus sailed the ocean blue!
Return
:b0:Columbus::
  MsgBox, Columbus set sail for the new world in 1492!
Return
```

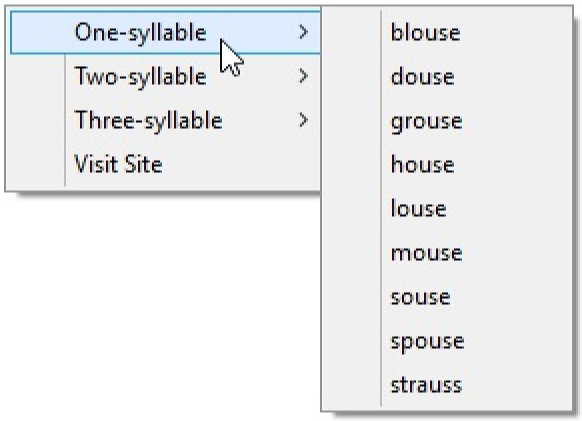
After loading the two Hotstrings (*1492* and *Columbus*), whenever typing either string followed by the space key or punctuation, the respective *MsgBox* windows pop up with the additional information.

The Web as a Reference

AutoHotkey has the capability to make any Web lookup easier. The short beginning AutoHotkey strip *SynonymLookup.ahk* demonstrates how to use the Windows Clipboard to quickly access reference sites on the Web. In this case, after hitting the Hotkey combination CTRL+ALT+L, *Thesaurus.com* opens in the default browser using the highlighted word in any text to find synonyms. (Referenced in the chapter "Why AutoHotkey for Writers, Bloggers, and Editors?") The ZIP file includes both the AHK script and the compiled EXE file which runs on any Windows PC—even without AutoHotkey installed.

More Word Choice

Specifically written for the poet inside us all, the Rhymes Pop-up script MenuRhymeMenu.ahk uses both Web lookup and the sample menu from the *OverusedWords.ahk* script to find rhymes. Simply highlight the target word (*house* in the example shown at the right), then use the CTRL+ALT+R Hotkey combination. AutoHotkey accesses the *http://www.rhymer.com* Web site and parses the list of soundalike words for addition to the pop-up menu.

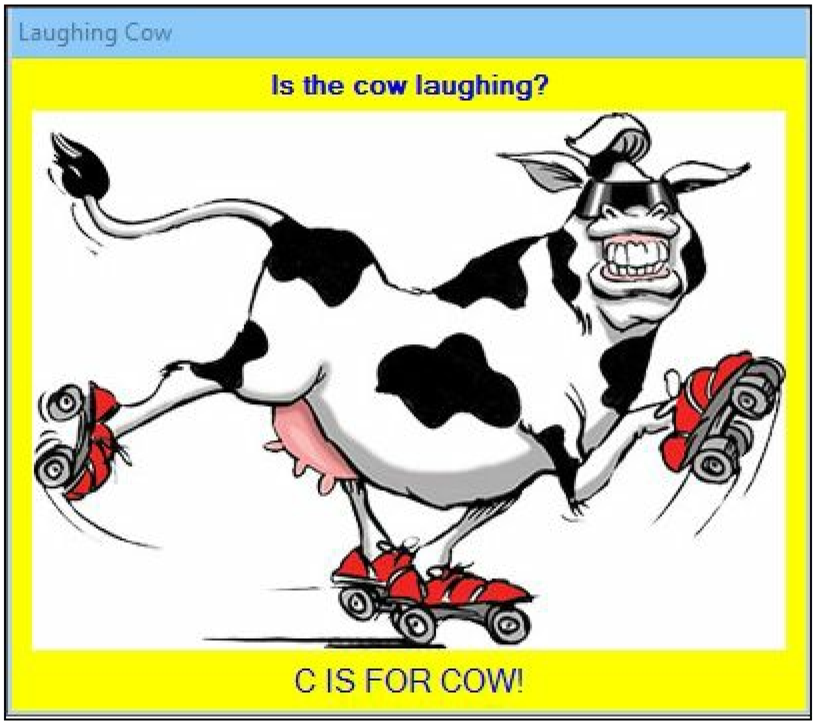


The PoeticWords.ahk script includes Hotstrings which instantly change five cent words into more exotic 50¢ permutations. The *PoeticWords.ahk* script contains over 500 Hotstrings for replacing common words with more pretentious counterparts.

These two scripts illustrate examples of how AutoHotkey might help both teachers and students in their quest for beautiful words (*PoeticWords.ahk*) and/or rhymes (*RhymeMenu.ahk*)—first introduced in "Why AutoHotkey for Poets?"

Early Childhood Education

AutoHotkey offers creative educational opportunities far beyond the manipulation of the English language. Simple multimedia scripts enhance learning for the younger student. As an example, the *NumbersSpeak.ahk* script includes the cow-skating.jpg (image file) and cow-madcow.wav (audio file) putting together an entertaining expression of sight and sound. Discussed in the blog "AutoHotkey Scan Codes, Speech, Sound, and Splash Images in Children's Apps."



The NumbersSpeak.ahk script uses the speaking SayWhat.ahk script as a basis for exploring techniques in writing children's educational software. The script employs the AutoHotkey SplashImage and SoundPlay commands, plus the ComObject function ComObjCreate("SAPI.SpVoice").Speak().

Press the letter "C", and the script displays the *SplashImage*, reads the words, then spells "cow", followed by the sound of a laughing mad cow mooing.

While the script is far from complete, it features some of the AutoHotkey tools available for building cute children's scripts. It is an apt companion for the educational TalkingText.ahk script which voices the letters of the keyboard. then makes animal sounds when certain animal names are spelled (e.g. cat, dog, bear, and more).

A Talking Keyboard

In the chapter "Why AutoHotkey for Grandparents?", I introduce a one-line AutoHotkey script which makes the keyboard voice the previously pressed key by pressing the SPACEBAR:

```
~Space::ComObjCreate("SAPI.SpVoice").Speak(A_PriorKey)
```

Load this script and let the little ones press keys, then the SPACEBAR to hear the key names spoken.

More Multimedia Presentations

The PhoneRing.ahk script (included in Chapter Six of the *AutoHotkey Applications* e-book) combines a graphic file, an audio file, and the computer voice to display a multimedia message. Activated with ALT+P, it demonstrates the use of a splash image with audio as a message, compiling all the files into one executable package which not only runs on any Windows computer but cleans up after itself.

Quiz Timing and Classroom Reminders

If you happen to keep a Windows computer on your classroom desk, other administrative tasks become easier. The EggTimer.ahk app script opens a window (after loading, press CTRL+F12) where you set the desired test time interval, then click Start. The time counts down to zero while a progress bar works its way from left to right. Once the countdown increments to zero, a bell rings, the computer voice says "Your eggs are ready!", and a window pops up with the same message. Now you have the perfect soft boiled egg (or, with a simple modification, maybe "Put your pencils down!"). Included in the AutoHotkey Applications e-book.

The Reminder.ahk script has gone through many iterations and has more to come—although I'm not sure when. I use it to tell me to take out the trash on Wednesday and for other quick temporary reminders. It is called up with the hotkey combination CRTL+WIN+R plus the most recent version has a Set Reminder option in the System Tray icon right-click menu.

In the classroom, it can remind the teacher when specific events must occur—such as a pop-quiz before the end of the period. The various versions of the Reminder files (some compiled into EXE files) have been included for people who are reading the book *Digging Deeper into AutoHotkey*. You can find those versions of the script in the *Reminder.zip* file.

Explanations of the *Appointment Reminder* script are incorporated in more detail in one section of the e-book *Digging Deeper Into AutoHotkey* and extended in Chapters Thirty-two and Thirty-three in the e-book *AutoHotkey Applications*. I encourage anyone to tailor the script for personal use—possibly adding multiple reminder capabilities and a repetitive reminder feature. I plan to do that myself when the time is right. (Maybe I should set up a reminder?)

Teach Programming with AutoHotkey

If you teach people to program, I would argue which language you use has less relevance than the techniques you teach. Learning to program is about learning how to think in a logical way while working with the tools common to most computer languages. For that reason, if I were instructing a class of budding system analysts, I would use AutoHotkey.

AutoHotkey includes all the tools (loops, if conditionals, etc.) that a programmer needs to use in any of the many programming languages. In fact, once you learn one language, you can quickly pick up another. It's merely a matter of learning the syntax and technical differences. AutoHotkey includes virtually everything that you find in other languages, plus some simple extra tools (Hotkeys and Hotstrings) offering immediate results for students. In addition, the easy construction of GUIs (Graphical User Interfaces) makes building powerful pop-up apps easy.

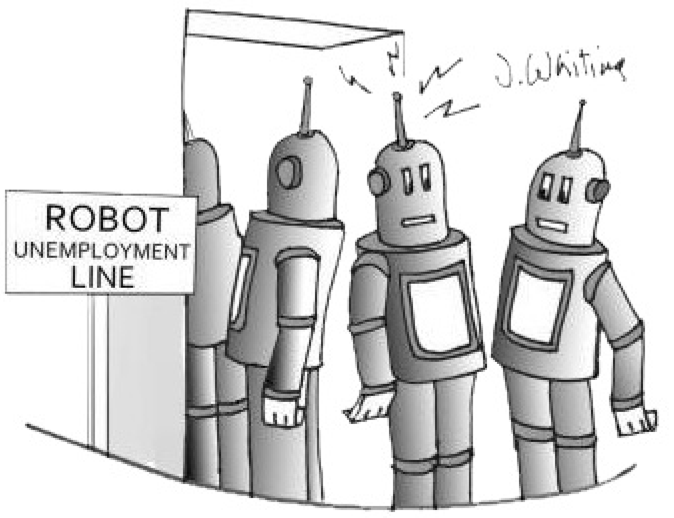
An AutoHotkey student gets instant useful results. The importance of immediate outcomes can't be overemphasized. Without them, demotivation soon follows. Most programming lessons start with the infamous "Hello, World!" routine. After that, the lessons become esoteric and metaphysical. Many may give up the journey while condemning programming as a boring occupation. However, when a student gets quick, practical results, the motivation level stays high.

Many more possibilities for AutoHotkey exist in education. Those mentioned here only represents my own minor endeavors. I have no doubt that creative teachers and professors will find numerous other ways to enrich the minds of their students with AutoHotkey on Windows computers.

Why AutoHotkey for Engineers and Scientists?

While Writing AutoHotkey Scripts Should Be No Problem for Most Engineers and Scientist, Many Might Be Surprised by How Much the Free Language Offers in Windows Tools

I'm not sure how many people with technical backgrounds are familiar with AutoHotkey. My guess is that quite a few have never heard of the free open source language. Without a personal referral or ubiquitous marketing, free software such as AutoHotkey often goes overlooked for a long period of time. It's not until individuals realize how much AutoHotkey can do for them that they start to explore the possibilities.



"I merely pointed out that he only knew Pi to 10 decimal places."

No software package does everything you want. That's why adding little extras makes any program better. The beauty of AutoHotkey is that in addition to automating individual Windows programs, it can cross boundaries and add more features to any Windows software. Plus, it has the capability to create special pop-up apps for specific usages. The Windows utility building features in AutoHotkey can be especially helpful for anyone working in a technical field.

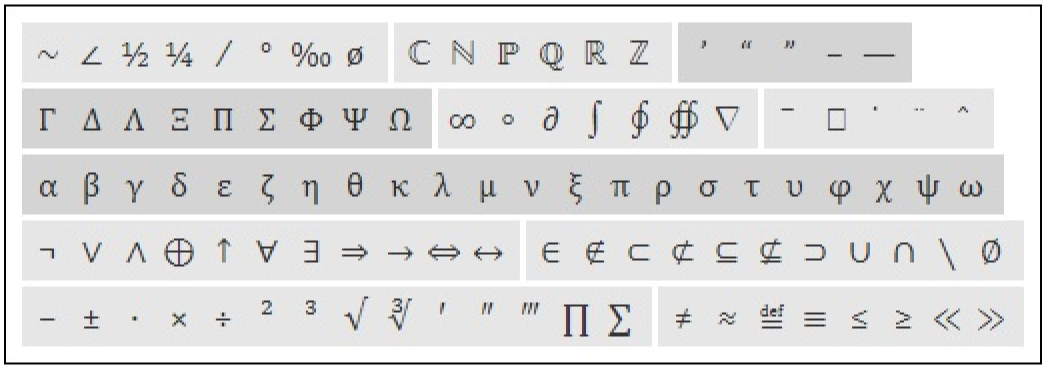
Scientific Terminology and Specialized Calculators

Engineers and scientists will discover many more applications for AutoHotkey than I introduce here. I confine my discussion to two roles for the Windows software: adding mathematical symbols to any documents and creating special purpose calculators and testers. Since most computer keyboards don't accommodate scientific notation, creating AutoHotkey Hotstrings comes in handy for inserting these characters and symbols into any Windows document or Web page. Plus, the advanced capabilities of AutoHotkey offer tools for building utility calculating and/or testing pop-up apps for specific areas of technical expertise.

Inserting Mathematical and Scientific Symbols

This first discussion outlines the adding of technical notations to any Windows document. Regardless of the engineering or science profession, writing reports, creating designs, and running experiments almost always requires the insertion of special characters and symbols into editing fields. While many word processing programs include special character tools for inserting unusual symbols and the built-in Windows CharMap (press WIN+r or +r, type *CharMap*, click *OK*) can make adding unique characters a little easier, AutoHotkey can set up universal Hotstrings for instantly inserting special characters into any Windows editing field without accessing extra menus or searching for the right symbol.

In addition to built-in special character software features and Windows CharMap, other alternatives exist to AutoHotkey Hotstrings. For example, you can access this convenient Web site (http://math.typeit.org/) to create your mathematical expressions. Yet, with AutoHotkey, you can turn each symbol into a Hotstring text replacement which works in any program without any Web lookup.


~ ∠ ½ ¼ / ° ‰ ø ℂ ℕ ℙ ℚ ℝ ℤ ' " " – —
Γ Δ Λ Ξ Π Σ Φ Ψ Ω ∞ ∘ ∂ ∫ ∮ ∯ ∇
α β γ δ ε ζ η θ κ λ μ ν ξ π ρ σ τ υ φ χ ψ ω
¬ ∨ ∧ ⊕ ↑ ∀ ∃ ⇒ → ⇔ ↔ ∈ ∉ ⊂ ⊄ ⊆ ⊈ ⊃ ∪ ∩ \ ∅
− ± · × ÷ ² ³ √ ∛ ′ ″ ‴ ∏ ∑ ≠ ≈ ≝ ≡ ≤ ≥ ≪ ≫
The TypeIt Web site offers these mathematical symbols for insertion into documents. With AutoHotkey, you can turn each one into a simple text expansion Hotkey which works in any Windows program.

AutoHotkey Hotstring Code

Writing simple one-line AutoHotkey Hotstrings makes each mathematical symbol available on any Windows keyboard. It's merely a matter of setting up Hotstring combinations which make the most sense to you. Here I offer a few possibilities:

```
:*:1/4*::¼     ; one-quarter
:*:1/2*::½     ; one-half
:*:3/4*::¾     ; three-quarters
:*?:deg*::°    ; degree
:*:<->::↔      ; two-direction-arrow
:*?:>->::→     ; right-arrow
:*?:<-<::←     ; left-arrow
:*?:<=>::⇔      ; double-two-direction-arrow
:*?:<=<::⇐     ; double-left-arrow
:*:>=>::⇒       ; double-right-arrow
:*:int*::∫     ; integral
:*?c:E1::¹     ; first power
:*?c:E2::²     ; squared
:*?c:E3::³     ; cubed
:*:sum*::∑     ; summation
:*?:+-::±      ; plus or minus
:*?:x*::×      ; multiplication sign
:*?:-|::÷      ; division sign
:*c:F*::ƒ      ; function
:*?:~~::≈      ; approximation
:*?:/=::≠      ; not equal
```

Embed each special character into the script by merely copying (highlight plus CTRL+C) from any source and pasting (CTRL+V) directly into the script editing program.

While not a complete list (I'll leave that to you), these Hotstrings demonstrate the possibilities. Each Hotstring combination establishes an activation sequence which is hopefully easy to remember.

Each AutoHotkey Hotstring consists of two sets of two colons (*::[activating characters]::[replacement text])*—all on one line. The beginning set of colons marks the start of the Hotstring, the second set of colons delimits the replacement characters or string. Each line acts as an independent command which can appear almost anywhere in a script—usually toward the end of the script after any setup code.

Options may appear between the first two colons. The * option between the two first two colons tells AutoHotkey to instantly replace the Hotstring without waiting for any other character or punctuation key press. The *?* option tells AutoHotkey to execute the replacement even if found in the middle of a word. The *c* option demands case sensitivity of the activating string.

Once loaded, by merely typing the Hotstring combinations, AutoHotkey returns the special characters as shown:

*int*F*(x)dx*

turns into:

∫ f '(x)dx

I suggest you pick activating Hotstrings which make the most sense to you. That way they will be easy to remember. Of course, you can always create a Hotkey pop-up window with the MsgBox command which reminds you of all the activating Hotstrings.

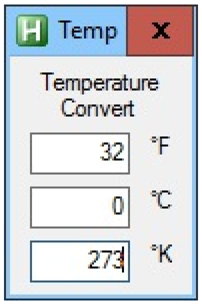
Special Purpose Calculators and Testers

Depending upon your field, you may need tools for calculating out-of-the-ordinary formulas. For some scenarios, you might use a spreadsheet, such as Microsoft Excel, or you may need to write a high-powered program for more complex testing, but wouldn't it be nice to use a few simple, quick tools for those unique calculations you need to make every day. You can build them with AutoHotkey.

To demonstrate how to quickly build pop-up tools with AutoHotkey, I wrote a short script for converting temperatures between Fahrenheit, Celsius, and Kelvin. (Since the current Windows 10 calculator offers the same function, I introduce this script for demonstration and educational purposes.)

Temperature Conversion Calculator

This *TempCalc.ahk* script creates a pop-up window which, depending upon what value changes, instantly converts degrees (°) Fahrenheit, Celsius, and Kelvin to each of the other scales. First, we set up the pop-up using the Gui (Graphical User Interface) command (image shown at right). Second, we embed the conversion formulas in subroutines called gLabels for calculating and updating the alternative scales.



Creating the AutoHotkey Pop-up Window

AutoHotkey contains commands for building pop-up windows. They include a wide variety of controls making them flexible and powerful. While this relatively routine example demonstrates one possibility, you'll find no limit on the number of ways a technical person can use GUI windows. (For a peek at the various GUI control types and their visual appearance, see "The Use and Images of AutoHotkey GUI Control Popup Windows.")

Since I've written extensively about the *Gui* command and its various options (particularly in the book *AutoHotkey Applications*, I only offer a brief overview of the commands and controls here. The following code sets up the pop-up window shown above:

```
Gui, -MaximizeBox -MinimizeBox

Gui, Add, Text, x15 Center ,Temperature`rConvert
Gui, Add, Edit, x10 vFar gFarChange Right w50, 32
Gui, Add, Text, x+10 , °F
Gui, Add, Edit, x10 vCel gCelChange Right w50, 0
Gui, Add, Text, x+10 , °C
Gui, Add, Edit, x10 vKel gKelChange Right w50, 273
Gui, Add, Text, x+10 , °K

Gui, Show, , Temp
Return
```

When adding (*Gui, Add,* ...) individual controls (text, edit fields, buttons, etc.), AutoHotkey GUI pop-up windows automatically resize themselves to fit. Each Gui, Add command inserts one control—by default placing them in the GUI window from the top-left corner to the bottom. Special positioning options, such as absolute location (e.g. *x10*) and relative placement (e.g. x+10), allow precise location of controls within the window. (See GUI positioning options.)

Created simultaneously with the control, the letter *v* appearing before a text name (e.g. *vFar*, *vCel*, and *vKel*) indicates the assignment of the control's variable. This variable holds the control's value.

If the control options list contains any text string preceded with the letter *g* (e.g. *gFarChange*, *gCelChange*, and *gKelChange*), then, depending upon the control, a Goto (*gLabel*) action subroutine activates when specific events occur. In the case of the Edit control, whenever detecting a change in the field contents (either by keyboard or AutoHotkey script action) the gLabel subroutine fires. Each subroutine label name (the label name followed by a single colon, e.g. *FarChange:)* which marks the beginning of the target subroutine must exist somewhere in the script when the file first loads.

Calculating Subroutines

AutoHotkey starts each subroutine with the label name (without the preceding *g*) followed by a single colon. Normally, a subroutine terminates with the Return command and can appear anywhere in the script after the initial setup code (i.e. in this case, they *cannot* appear within all the *Gui* command code lines). The following *gLabels* carry out the calculations needed as each temperature edit field changes, respectively:

```
FarChange:          ; Fahrenheit change
  ControlGetFocus, OutputVar
  if (OutputVar = "Edit1") {
     Gui, Submit, Nohide
     Cel := (Far - 32)*5/9
     GuiControl, ,Cel, % Round(Cel,0)
     Kel := Cel + 273.16
     GuiControl, ,Kel, % Round(Kel,0)
  }
Return
```

```
CelChange:       ; Celsius change
  ControlGetFocus, OutputVar
  if (OutputVar = "Edit2") {
     Gui, Submit, Nohide
     Far := (Cel*9/5)+32
     GuiControl, ,Far, % Round(Far,0)
     Kel := Cel + 273.16
     GuiControl, ,Kel, % Round(Kel,0)
  }
Return

KelChange:        ; Kelvin change
  ControlGetFocus, OutputVar
  if (OutputVar = "Edit3") {
     Gui, Submit, Nohide
     Cel := Kel - 273.16
     GuiControl, ,Cel, % Round(Cel,0)
     Far := (Cel*9/5)+32
     GuiControl, ,Far, % Round(Far,0)
  }
Return
```

A few features of note appear in each of the three *gLabel* subroutines above. First, the ControlGetFocus command grabs the name of the active control. AutoHotkey uses this control name to isolate any action with an IF conditional whenever that control has focus. Otherwise, since any change in an edit field (by keyboard or by the script) activates its respective *gLabel*, all three subroutines would continually fire as they change the other control values.

The routines require the Gui, Submit command to capture the new value of the temperature in the target field.

After calculating the new temperatures, each subroutine uses the GuiControl command to update the other temperature fields. The powerful forced expression operator % immediate evaluates and inserts the value of the Round() function.

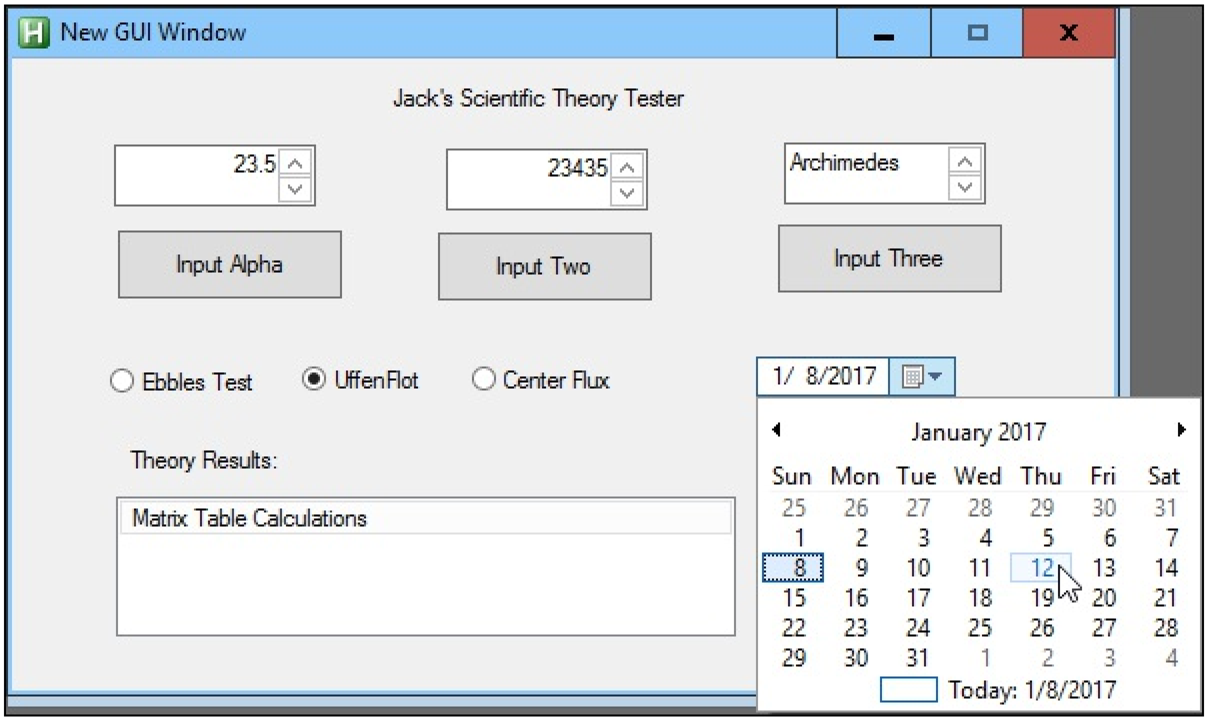
Add the following Hotkey routine to the end of the script to recall the *TempCalc.ahk* window at any time:

```
#!t::
  Gui, Show
Return
```

Press WIN(⊞)+ALT+T to reactivate (*Gui, Show*) the pop-up.

Power Theory/Testing Tools

The simple temperature conversion calculator may not convince an engineer or scientist of the usefulness of AutoHotkey, but take a look at this next Theory Tester:



Jack's Scientific Theory Tester

23.5

23435

Archimedes

Ebbles Test [✓ UffenFlot] Center Flux

1/ 8/2017

Theory Results:

Matrix Table Calculations

January 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Today: 1/8/2017

Of course, this AutoHotkey pop-up window was fabricated merely for the purpose of demonstrating how a really useful theory testing tool might operate. While it doesn't actually do anything, it looks important and maybe even scientific in nature. I leave it to you to make it work.

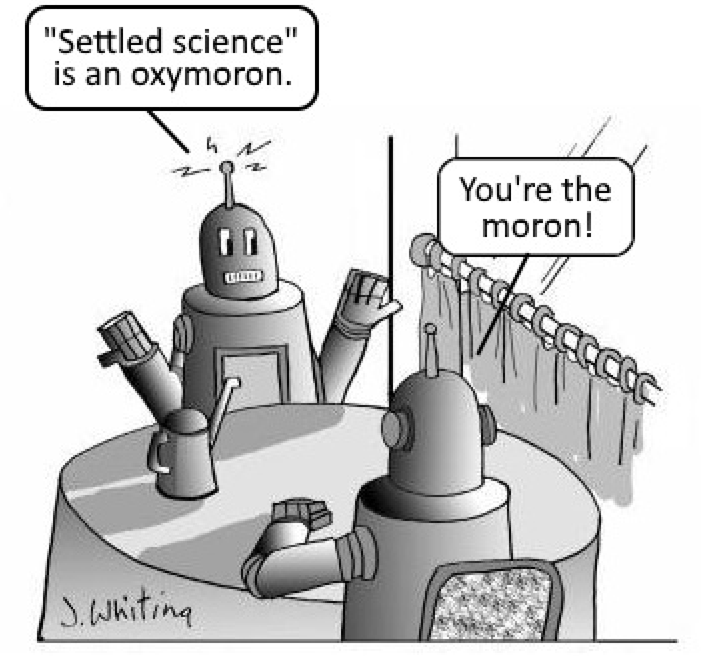
Okay, maybe this one is a total fiction. (I built the pop-up with Rajat's AutoHotkey SmartGUI Creator, but it is non-functional.) I can only guess how someone might implement these techniques in engineering or science. The work is so varied that it's up to the AutoHotkey user to determine how to best build the most useful tools.

Why AutoHotkey for Internet Trolls?

If You Plan on Being One of the Most Annoying People on the Web, Why Not Make It Easy on Yourself?

Note: If you're an Internet troll, please don't take offense at anything I say here. I'm merely showing how AutoHotkey makes trolling easier—as the free software does with anything you do on any Windows computer. Not that trolls need any help—other than psychological.

Internet trolls patrol cyberspace in an effort to right the wrongs perpetrated by unsuspecting users…or, maybe, they just want to make themselves feel better by making others feel worse. Whatever! The important point is that even Internet trolls can make good use of the free AutoHotkey tools available for their Windows computers.



Trolling Robot

Some people think that AutoHotkey software should only be used for good, but if you like to harass people on the Web, right or wrong, AutoHotkey may be the tool for you. Internet trolls will be surprised at how easy AutoHotkey makes harassing people.

Disclaimer: Don't blame AutoHotkey for this blog. Any tool can be used for good or evil. While a hammer can build a house, it can also tear it down.

(If you're new to AutoHotkey, please see this "Introduction to AutoHotkey: A Review and Guide for Beginners.")

Yell at Your Trolling Target

One of the most important forms of trolling involves expressing everything in all capital letters. In digital communications, people equate all caps with yelling—but a troll knows that using uppercase letters merely represent a way to emphasize a point. Therefore, the CapsLock key should always be locked in the ON position. The problem arises when you either forget to turn CapsLock on or you accidentally turn it off. Fortunately, if you install and load AutoHotkey, a simple one-line script solves the problem:

```
SetCapsLockState, AlwaysOn
```

The SetCapsLockState command gives you absolute control over your CapsLock key. (This may be one of the few areas of a troll's life where it can establish any level of control.) However, as unlikely as it may be, there could be times when you don't want the CapsLock key ON. In that case, you might use an AutoHotkey technique for creating a key combination which turns any highlighted text into all caps after the fact:

```
^u::    ; capitalize text
  OldClipboard:= ClipboardAll
  Clipboard:= ""
  Send, ^c ;copies selected text
  ClipWait 0
  If ErrorLevel ; for when you forget to select text
  {
    MsgBox, No Text Selected!
    Return
  }
  StringUpper Clipboard, Clipboard
  Send %Clipboard%
  Sleep 100
  Clipboard:= OldClipboard
Return
```

Okay, while this may look a little complicated for the average troll, it's not that difficult. Using what I call the "Standard AutoHotkey Windows Clipboard Routine", selected text (highlighted by dragging the mouse cursor over the text with the left mouse button held down) gets copied to the Windows Clipboard, changes the text to all uppercase letters with the StringUpper command, then pastes it back in place of the original text—all without altering the previous contents of the Windows Clipboard. This approach adds more flexibility to your trolling activities while maintaining the option to go all caps anytime.

Don't Use AutoHotkey AutoCorrect!

Whether from typos or pure ignorance, nothing adds Net cred to your trolling activities like misspelled words and grammar errors. What you have to say is far too important to care about trivialities such as accepted spelling and punctuation promoted by the establishment. Just let the words flow. That's why you absolutely should not run the AutoHotkey AutoCorrect script which fixes thousands of commonly misspelled English words on the fly.

However, you also should not deliberately add misspellings or insert faulty grammar. Just let the words randomly fall wherever. (You may want to turn off your copy of the free version of Grammarly. *Full Disclosure: I get 20¢ if you install the free version of Grammarly using this link.*) Intentional errors are a form of troll cheating and regarded as an attempt to disguise oneself—which, come to think of it, may not be a bad idea.

Disguise Yourself as a Candian

Many Internet trolls hide behind pseudonyms to prevent their identification—at least, without a warrant. However, even with a fake name, if you don't take further steps, people may recognize you as someone they know. Throw them completely off track by pretending to be from another country. If I were a troll (which I'm not), I would pick Canada because most people from the United States can easily be fooled by simple Canadian text misdirection. (Warning: Don't try this on real Canadians because they will immediately recognize you as a phony and a scoundrel, eh?)

The simplest way to convince someone that you're Canadian is to add "eh?" to the end of your sentences. This immediately alerts those of us from south of the border that we're talking to one of our brethren from the north. While we might use “Eh?” when wondering what the heck someone’s talking about, many Canadians use it as a unique way of emphasizing their last statement. With AutoHotkey you can ensure that you add it to your rants with the following line of code:

```
::`.::, eh?
```

After running this one line script, whenever typing a period followed by a space, AutoHotkey replaces it with “, eh?” and a space. Alas, this Hotstring replaces every occurrence of a period with the comma followed by “eh?” That’s a bit too much. The following Hotstring routine only replaces every third occurrence of the period.

```
Count = 0  ; in auto-execute section of the script

:B0:`.::
  Count++
  If (Count = 3)
   {
   Send {BS 2}, eh?
   Count = 0
   }
Return
```

For information on how Hotstrings work, see AutoHotkey Hotstrings at the main AutoHotkey Web site. (For even more detailed discussion, see my book *Beginning AutoHotkey Hotstrings*.)

Of course, if you understand where it’s appropriate to append “eh?” to the end of a sentence, then, with AutoHotkey, you can create your own Hotkey keyboard shortcut:

```
^!E::Send `, eh?  ; activates with CTRL+ALT+E
```

Then, anytime you want to insert the Canadian add-on into your Windows documents or Web browser, simultaneously press CTRL+ALT+E (represented in the AutoHotkey code by *^!E*). (Learn more about Hotkeys at https://autohotkey.com/docs/Hotkeys.htm or check out my book AutoHotkey Hotkey Techniques.)

While much more subtle than the “eh?”, adding the following Hotstring to your standard AutoHotkey script may assist you in duping an American target of your trolling:

```
::about::aboot
```

While typing on your keyboard, this Hotstring automatically replaces every occurrence of “about” with the “aboot” Canadian pseudo-homonym. This immediately makes people from the United State think you’re a Canadian. Of course, Canadians won’t be fooled because “aboot” doesn’t appear in their vocabulary. Although they might pronounce the word “aboat”, they still spell it “about”—the same as every other English speaking person. Admittedly, since you can’t hear dialects when reading on the Web, this may seem like a pretty silly suggestion for such a serious blog. It might be better to create your own Canadian AutoCorrect script.

Canadian AutoHotkey AutoCorrect

If you really want to make someone think that you’re a Canadian, then use AutoHotkey to build your own specialized AutoCorrect script altering American spellings into the Canadian counterparts. (Caution: You can’t use a British AutoCorrect script!) By researching the differences between American and Canadian spelling you can write a script which (with any luck) will make you sound (in written text) distinctly

Canadian. For example, the script may include the following:

```
::color::colour
::center::centre
::defense::defence
::offense::offence
::practice::practise
::traveled::travelled
```

Then, throw in a few “eh?”‘s and you have it made.

This blog barely scratches the surface of what Internet trolls can do with AutoHotkey. Of course, if you don’t use Windows, then you’ll need to find another way to do it.

Disclaimer: Some of my best friends are Canadians, although I don’t hear their accent anymore. Maybe they lost it?

* * *

Note: Nothing in this chapter should be taken as any type of bias against anyone who spends days and/or nights trolling on the Web. Trolls are people too! After all, when you have no life, what else are you suppose to do? Everyone needs a hobby!

E-Books for Making Your Journey into AutoHotkey Scripting Easier

"Every programmer has moments where he or she achieves a breakthrough and their code actually works. Their first impulse is to tell someone…anyone, 'Look what I just did!' Alas, often there is no one to show their accomplishment. At least, no one who either understands or cares about their success. Maybe that's why I write about my AutoHotkey journeys. I need to tell someone when (ironically?) some code I wrote does what it's supposed to do."

"I think my primary goal is to entice other people into learning AutoHotkey. Not just because it's a powerful, simple Windows utility language, nor because writing working scripts is extremely satisfying (and fun), but I want more people using and understanding AutoHotkey so that I can tell them whenever I manage to eke out a minor AutoHotkey achievement."

"Although a number of ideas in my books could be useful to experienced programmers, the bulk of my work aims at novices who want to learn the ins and outs of AutoHotkey scripts. If you already know how to write scripts, then these books may only act as a source of ideas. For the knowledgeable coder, the basic AutoHotkey commands and features are available on the Web at AutoHotkey.com *and you can teach the language to yourself—as I did."*

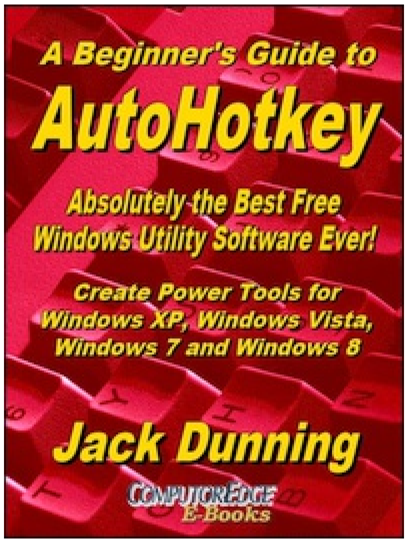
"However, I do admit that I often refer back to my own books while writing this AutoHotkey blog, because, not even I can remember everything that I wrote."

— Jack Dunning

* * *

New to AutoHotkey? See "Introduction to AutoHotkey: A Review and Guide for Beginners."

* * *


A Beginner's Guide to
AutoHotkey
Absolutely the Best Free Windows Utility Software Ever!
Create Power Tools for Windows XP, Windows Vista, Windows 7 and Windows 8
Jack Dunning
ComputorEdge E-Books

This third edition of the e-book which contains more chapters and an index to the AutoHotkey commands is available in e-book format at the ComputorEdgeBooks Web site linked below). Jack's *A Beginner's Guide to AutoHotkey, Absolutely the Best Free Windows Utility Software Ever!: Create Power Tools for Windows XP, Windows Vista, Windows 7, Windows 8 and Windows 10* (preferred, EPUB format for iPad, Android, and computers; MOBI; and PDF for printing) offers a gentle approach to learning AutoHotkey.

For more details about *A Beginner's Guide to AutoHotkey* (Table of Contents and the entire book index), click here!

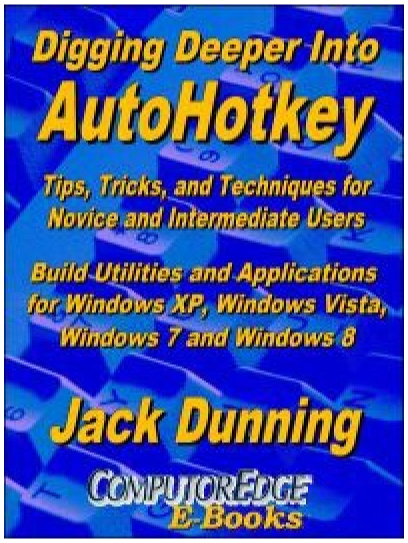
Building Power Tools for Windows XP, Windows Vista, Windows 7, Windows 8 and Windows 10, AutoHotkey is the most powerful, flexible, *free* Windows utility software available. Anyone can instantly add more of the functions that they want in all of their Windows programs, whether installed on their computer or while working on the Web. AutoHotkey has a universality not found in any other Windows utility—free or paid.

Originally based upon the series of articles in *ComputorEdge*, Jack takes you through his learning experience as he explores writing simple AutoHotkey scripts for adding repetitive text in any program or on the Web, running programs with special hotkeys or gadgets, manipulating the size and screen location of windows, making any window always-on-top, copying and moving files, and much more. Each chapter builds on the previous chapters.

For an EPUB (iPad, NOOK, etc.) version of A Beginner's Guide to AutoHotkey click here!

For a PDF version for printing on letter size paper for inclusion in a standard notebook of A Beginner's Guide to AutoHotkey click here!

* * *


Digging Deeper Into
AutoHotkey
Tips, Tricks, and Techniques for Novice and Intermediate Users
Build Utilities and Applications for Windows XP, Windows Vista, Windows 7 and Windows 8
Jack Dunning
ComputorEdge E-Books

Jack's second AutoHotkey book, *Digging Deeper Into AutoHotkey* (preferred, EPUB format for iPad, Android, and computers; MOBI; and PDF for printing) is comprised of updated, reorganized and indexed columns from *ComputorEdge* is now available. Since the columns were not all written in a linear fashion, the book has been reorganized and broken up into parts by topic. The book is not for the complete beginner since it builds on the information in *A Beginner's Guide to AutoHotkey*.

For more details about *Digging Deeper into AutoHotkey* (Table of Contents and the entire book index), click here!

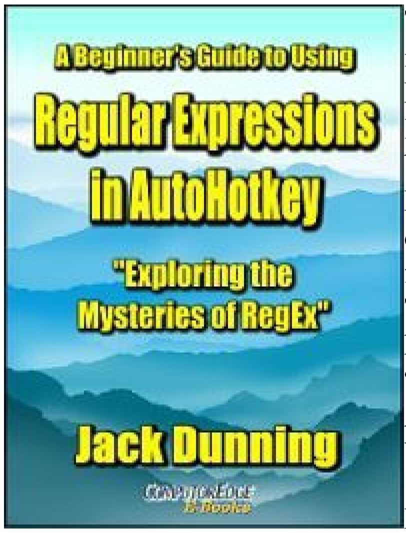
However, if a person is reasonably computer literate, they could go directly to this book for ideas and techniques without the first book.

For an EPUB (iPad, NOOK, etc.) version of Digging Deeper into AutoHotkey click here!

For a PDF version for printing on letter size paper for inclusion in a standard notebook of Digging Deeper into
AutoHotkey click here!

* * *

* * *


A Beginner's Guide to Using
Regular Expressions in AutoHotkey
"Exploring the Mysteries of RegEx"
Jack Dunning
ComputorEdge E-Books

This *Beginner's Guide to Using Regular Expressions in AutoHotkey* is not a beginning level AutoHotkey book, but an introduction to using Regular Expressions in AutoHotkey (or most other programming languages). To get the most from this book you should already have a basic understanding of AutoHotkey (or another programming language). Regular Expressions (RegEx) are a powerful way to search and alter documents without the limitations of most of the standard matching functions. At first, the use of RegEx can be confusing and mysterious. This book clears up the confusion with easy analogies for understanding how RegEx works and examples of practical AutoHotkey applications. "Regular Expressions in AutoHotkey" will take you to the next level in AutoHotkey scripting while adding more flexibility and power to your Windows apps.

For more details about Regular Expressions in AutoHotkey (Table of Contents and the entire book index), click here!

Table of Contents

Why AutoHotkey? 2

How Free AutoHotkey Adds Power to Your Windows PC and Provides You Important Brain Food 2

Ideas and Applications for How AutoHotkey Can Aid Various Occupations and Individuals in their Daily Computing Needs 2

by Jack Dunning 2

Why AutoHotkey for Every Windows Computer? 6

If, after reading this book, you don't come to the conclusion that you should make AutoHotkey part of your computing life, then you don't use a Windows computer. Maybe all you do is add one two-letter Hotstring which expands into your e-mail address. That alone will save you tons of time. 8

More AutoHotkey References 8

Why AutoHotkey for Learning to Program? Windows Programming Made Easy! 9

Never Programmed Before? You'll Find No Better Language for Learning Than AutoHotkey! Plus, It May Save Your Intellect from Precipitous Decline. 9

Languages for Learning to Program 9

AutoHotkey Gives Novices Instant Results 10

AutoHotkey Hotkeys 10

AutoHotkey Hotstrings for Text Expansion and Correction 10

Do As Little Or As Much As You Like 11

AutoHotkey Is an Easy-to-Read Language 11

Forgiving Structure and Syntax 11

Everything You Need Is on the Web…Mostly 12

Train Your Brain 12

Lifetime Benefits of Learning to Program 13

Why AutoHotkey for Writers, Bloggers, and Editors? 14

If You Write or Edit For a Living (or Fun) and Use a Windows Computer (Most People Do), Then You Should Use the Free AutoHotkey Software 14

AutoHotkey Works Everywhere 14

AutoHotkey Hotstrings 14

AutoCorrect—The Number One Reason for Writers to Use AutoHotkey 15

Free AutoHotkey Scripts and Apps 15

Scratching the Surface 17

Why AutoHotkey for Artists and Graphic Designers? 19

While Using AutoHotkey with Windows Makes Sense for Most Professions, Its Value Is Not So Obvious for Artists and Graphic Designers 19

Adding Enhancements to Your Graphic Tools 20

Empowering Windows Paint with AutoHotkey 20

Why AutoHotkey for Poets? 24

Erstwhile Multifarious Poets Optated for Quill and Parchment. Forthwith, AutoHotkey Propounds the Furtherance of Lyrical Ruminations on Windows Computers. 24

Beautiful Hotstrings 24

Rhymes from the Web 25

Why AutoHotkey for Chefs and Dieticians? 27

While You Don't Often See Television Chefs Using Computers, AutoHotkey Offers Guidance When Cooking "Jack Stuffed Cheeseburgers" at Home 27

Jack Stuffed Cheeseburgers 28

More Nutrition and Recipe AutoHotkey Scripts 30

The Egg Timer 30

Counting Calories 30

Recipe Book 31

Why AutoHotkey for Grandparents? 33

Remember All Your Grand Kid's Birthdays and Their Ages! There's No Limit to the Number of Ways You Can Amuse Your Grandchildren with AutoHotkey, Plus It Gives Your Brain a Much Needed Workout! 33

Remembering Birthdays and Ages 33

A Simple One-Line Script to Teach the Computer Keyboard and Alphabet 34

More AutoHotkey Entertainment and Education 35

Exercise the Brain 36

Why AutoHotkey for Students? 38

Ten Reasons Why Every Student Should Use the Free AutoHotkey Windows Tools! AutoHotkey Helps You So Much Academically That You'll Think You're Cheating! Plus, It's Delicious Brain Food and Helps Get You a Job! 38

Ten Reasons Students Should Use AutoHotkey 38

Reason One: AutoCorrect on the Fly the Most Commonly Misspelled English Language Words 39

Reason Two: Write Gender Neutral Essays and Term Papers 40

Reason Three: Insert Funky Foreign Language Characters 41

Reason Four: Instantly Add the Date to Any Document 41

Reason Five: Collect References for Research Papers 42

Reason Six: Protect Yourself from Lost Work 43

Reason Seven: Save Typing Time with Instant Hotkeys 44

Reason Eight: Eliminate Student's Most Commonly Overused Words in Papers 45

Reason Nine: Quickly Find Even More Synonyms 45

Reason Ten: Improve Your Mind and Get a Job 45

Why AutoHotkey for Teachers and Educators? 48

If You and Your Students Have Access to Windows Computers, Then You're Set! If Not, Well… 48

AutoCorrect That Teaches 48

The Web as a Reference 49

More Word Choice 49
Early Childhood Education 50
A Talking Keyboard 50
More Multimedia Presentations 50
Quiz Timing and Classroom Reminders 51
Teach Programming with AutoHotkey 51
Why AutoHotkey for Engineers and Scientists? 53
While Writing AutoHotkey Scripts Should Be No Problem for Most Engineers and Scientist, Many Might Be Surprised by How Much the Free Language Offers in Windows Tools 53
Scientific Terminology and Specialized Calculators 53
Inserting Mathematical and Scientific Symbols 53
AutoHotkey Hotstring Code 54
Special Purpose Calculators and Testers 55
Temperature Conversion Calculator 55
Creating the AutoHotkey Pop-up Window 55
Calculating Subroutines 56
Power Theory/Testing Tools 57
Why AutoHotkey for Internet Trolls? 59
If You Plan on Being One of the Most Annoying People on the Web, Why Not Make It Easy on Yourself? 59
Yell at Your Trolling Target 59
Don’t Use AutoHotkey AutoCorrect! 60
Disguise Yourself as a Candian 60
Canadian AutoHotkey AutoCorrect 61
E-Books for Making Your Journey into AutoHotkey Scripting Easier 63