

AutoHotKey: an Editor-independent Alternative to SAS® Keyboard Abbreviations

Shane Rosanbalm, Rho, Inc.

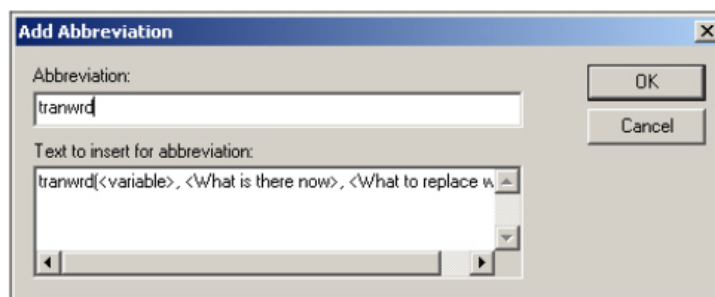
ABSTRACT

Have you ever been editing a SAS program in a text editor (UltraEdit, NotePad++, etc.) and found yourself thinking, "I wish I had access to my SAS keyboard abbreviations"? AutoHotKey (AHK) is free open-source macro creation and automation software for Windows that allows users to automate repetitive tasks. It's a lot like SAS keyboard abbreviations, but with one major perk: not only does AHK work in your SAS editor, but it also works in text editors, word processors, web browsers, and just about any other application you can think of. If you like SAS keyboard abbreviations but wish they worked in other applications, AHK is the missing piece of software you've been yearning for. In this paper you will learn how to install AHK and get started creating your own editor-independent abbreviations.

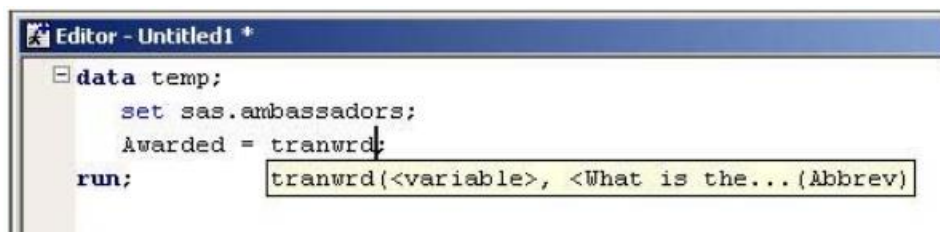
INTRODUCTION

The concept of keyboard abbreviations has been around for many years and has been the topic of many a conference paper. For those unfamiliar with keyboard abbreviations, here is the briefest of introductions.

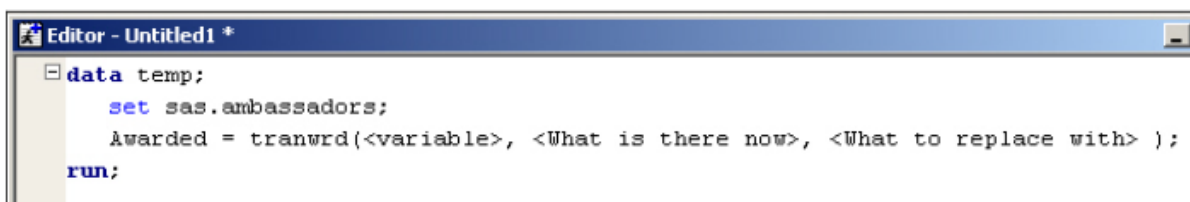
You begin by opening the Enhanced Editor and pressing Ctrl+Shift+A. This brings up a window with two fields:



The next time you type 'tranwrd' in the Enhanced Editor you will see a light yellow box below your cursor. This yellow box is essentially a question: "do you want to replace this text with the abbreviation you defined earlier?"



If your next keystroke is Enter, the editor will replace the abbreviation with the "Text to insert for abbreviation".



I have used keyboard abbreviations for many years and have always been a big fan. But, keyboard abbreviations are far from perfect. Among the stumbling blocks that I've encountered are:

- The "Text to insert for abbreviation" field is small and the font is proportional, making it hard to get longer abbreviations to look right on the first try.
- Sharing abbreviations requires an awkward export/import process, and there is not a convenient way to export/import a subset of abbreviations.

- It is easy to lose your abbreviations. For instance, when you get a new PC, or you get a new version of SAS installed on your current PC, or your organization switches from Display Manager (DM) to Enterprise Guide (EG).
- And most importantly, the abbreviations you create in SAS are not available to you in other editors.

Again, I'm a big fan of SAS keyboard abbreviations. But these stumbling blocks have always left me wishing that there were a better way. And then one day I happened upon a piece of software called AutoHotKey (AHK).

AUTOHOTKEY BASICS

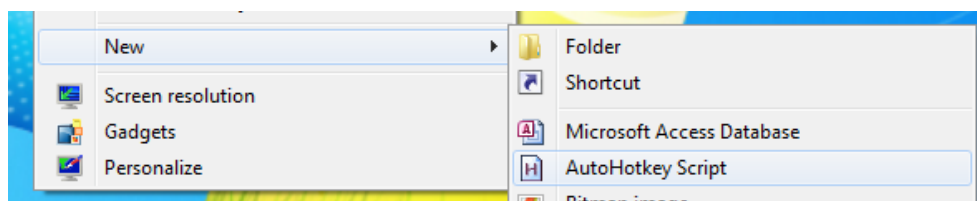
AHK is free open-source macro creation and automation software for Windows that allows users to automate repetitive tasks. In AHK there is the concept of a "hotstring". It is just like a keyboard abbreviation in SAS except that it is available in virtually every application that you have on your PC. It works in the various SAS editors (DM, EG, and Studio), word processors (MS Word, Libre Office Writer, etc.), text editors (TextPad, UltraEdit, NotePad++, etc.), web browsers (Chrome, FireFox, Explorer, etc.), and just about any other application that you can think of. In the following sections I will walk you through how to install AHK and get started using hotstrings.

Installing AHK

Download AHK at autohotkey.com using the Download link on the right side of the page. Choose Exit once the installer completes.

Creating Your First .ahk File

To create your first .ahk file simply right-click anywhere on the Windows desktop and select **New -> AutoHotKey Script**.



Open the resulting .ahk file using any old text editor. You will see that several lines' worth of options and comments have been pre-specified. Keep these as they are.

```

New AutoHotKey Script.ahk x
1 ;
2 ; AutoHotkey Version: 1.x
3 ; Language:      English
4 ; Platform:      Win9x/NT
5 ; Author:        A.N.Other <myemail@nowhere.com>
6 ;
7 ; Script Function:
8 ; Template script (you can customize this template by editing "ShellNew\Template.ahk" in your Wi
9 ;
10
11 #NoEnv ; Recommended for performance and compatibility with future AutoHotkey releases.
12 SendMode Input ; Recommended for new scripts due to its superior speed and reliability.
13 SetWorkingDir %A_ScriptDir% ; Ensures a consistent starting directory.
14

```

Creating Your First Hotstring

We might choose to define the Hotstring 'imho' and associate with that the longer text 'in my humble opinion'. The AHK syntax for this is relatively straightforward.

```

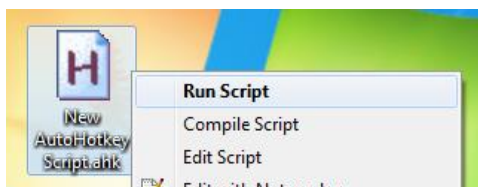
14
15 ::imho::in my humble opinion
16

```

Once you have added this line of code as the last line of your .ahk file, save the file.

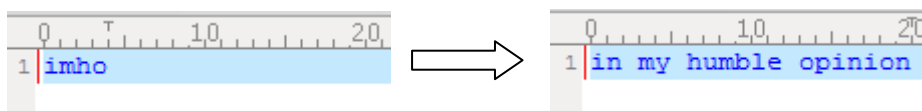
Launching Scripts

In order to be able to use the .ahk script that you have just created, right-click on the .ahk file and select **Run Script**.



Using Your First Hotstring

Open any application you like (UltraEdit is used in the screen shots below). Type `imho`, press the Enter key, and watch the text-substitution magic happen!



Creating your Second Hotstring

In the same .ahk file, let's create a second hotstring. But let's make this one a little more SAS-centric. I happen to like adding a lot of dashes around my SAS comments. Something like:

```
*-----;
*----- Dashing off a comment! -----;
*-----;
```

The AHK syntax for multi-line text substitution requires the addition of parentheses around the text to insert.

```
::commentblock::
(
*-----;
*-----;
*-----;
)
```

Re-launching Your Script

Now that we've typed a second hotstring into our script we have to go through the launch procedure again (save, right-click, Run Script). The hotstring `commentblock` is now available.

ADVANTAGES OF AUTOHOTKEY

There are many advantages to using AHK in place of SAS keyboard abbreviations.

- In the PC SAS (i.e., DM) Enhanced Editor you are stuck with a very small field in which to type your text to insert. Moreover, the font is proportional. This makes editing an abbreviation quite difficult. EG has a better abbreviations editor (monospace font and a re-sizeable window), but in my experience it's still not quite as convenient as managing abbreviations/hotstrings in a text editor. And as of this writing, Studio does not appear to support keyboard-based abbreviations.
- The process of sharing abbreviations/hotstrings is greatly simplified. Since the AHK hotstrings are stored as text, there is no awkward import/export process. Moreover, if someone gives you their complete set of 60 hotstrings, you can easily skim through them in a text editor and only transfer the ones that you're interested in to your own AHK script. It's a simple copy/paste/re-launch operation.
- Losing AHK hotstrings is a lot harder than losing SAS abbreviations. As long as you store your .ahk file in a location with regular network backups, you need never fear losing your hotstrings when you get a new PC, or a new version of SAS, or switch to EG, or whatever.
- And most importantly, the AHK hotstrings are available to you in any editor. Do you prefer writing code in TextPad and submitting in batch? Are you wanting to experiment with Studio but are afraid of the productivity hit that you'll take from losing your beloved keyboard abbreviations? Can you think of applications for hotstrings outside of SAS (sentences/paragraphs that you frequently write in emails, hard-to-type information in various web forms, etc.)? All of these are great reasons to give AHK a try.

DISADVANTAGES OF AUTOHOTKEY

While I adore AHK, it is not without its shortcomings too.

- The first real disadvantage of AHK is that the script has to be re-launched each time you log in to your computer. Fortunately, the launch process can be automated. Simply create a shortcut for your .ahk file (right-click, Create shortcut) and move the shortcut to your Windows Startup folder. Once you have placed the shortcut in the Windows Startup folder, the script will automatically launch every time you start your PC.
 - o In Windows 7 the Startup folder is: `C:\Users\<user name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`.
- The second disadvantage is that AHK hotstrings do not resolve nearly as quickly as SAS keyboard abbreviations. For instance, if you have a 300-character abbreviation stored in EG, the substitution is nearly instantaneous. Whereas if you use AHK, the same substitution will take several seconds (AHK inserts one character at a time). If you are an extremely proficient typist, this speed difference may be frustrating. One possible workaround to this speed difference would be to use both AHK and SAS, with slightly different hotstrings/abbreviations in the two applications. For instance,
 - o In AHK define `longstring`.
 - o In EG define `longstringEG`.
 - o When you are using EG, type `longstringEG` for faster substitution. And when you are using another editor, type `longstring`.

ADDITIONAL AUTOHOTKEY TOPICS

Comments

In AHK you begin your comments with a semicolon. Just like in SAS, it is a good idea to comment your AHK scripts.

```
; This is a comment in AHK.
```

AHK assumes that each statement/command is contained on a single line. So, if you want a multi-line comment, you will need to use multiple semicolons.

```
; This is a multi-line comment
; in AHK.
```

Resolution

By default Hotstrings resolve when the first post-Hotstring keystroke is any of the following:

```
-()[]{}:;'"/\, .?!`n`t
```

Some programmers prefer to limit the set of post-Hotstring resolution keystrokes by adding the following option to the .ahk file:

```
#Hotstring EndChars `n`t
```

This option only allows Hotstrings to resolve if the first post-Hotstring keystroke is either the Tab or Enter key.

Escape Characters

Suppose you wanted to turn the following into a hotstring.

```
%dupcheck
  (data=
    ,var=
  )
```

The naïve first attempt would be to add the following lines to the .ahk script.

```

::dupcheck::
(
%dupcheck
  (data=
    ,var=
  )
)

```

Unfortunately, this does not work. Having begun our multi-line hotstring with the left parenthesis, AHK is now looking for a line that begins with a right parenthesis as a sign that the hotstring has ended. We have two lines that begin with a right parenthesis, which confuses AHK and causes an error.

Fortunately, AHK has the concept of escape characters. If you wish to have the first right parenthesis treated as an ordinary character (and not the end of a hotstring), simply put a tick mark in front of it.

```

::dupcheck::
(
%dupcheck
  (data=
    ,var=
    `)
)

```

Hotkeys (aka, Keyboard Macros)

SAS has keyboard macros, and AHK supports similar functionality. A full explanation of this AHK functionality is beyond the scope of this paper, but perhaps one small example will give you a sense of what is possible in AHK.

Suppose you want to define a keyboard macro that will insert the current date in YYYY-MM-DD format. In AHK the syntax for a keyboard macro is slightly different than for a hotstring. You do not start with two colons, and you use the keywords `send` and `Return`. Here is an interesting application of keyboard macros in AHK:

```

^!d::
    send, %A_YYYY%-A_MM%-A_DD%
Return

```

In AHK `^` corresponds to the Control key and `!` corresponds to the Alt key. Therefore `^!d` translates to Ctrl+Alt+d. So, if you add these lines to your script, then every time you press Ctrl+Alt+d the current date will be displayed in YYYY-MM-DD format.

A GITHUB REPOSITORY

Prior to writing this paper, I wanted to share AHK with my coworkers. To do so I created a repository on GitHub, the free online code-sharing site. A modest collection of hotstrings is available in the `sas` folder.

<https://github.com/srosanba/sas-autohotkey>

The hotstrings are grouped into separate files for ease of browsing. Most readers will probably find the file `sas-comments.ahk` to be the most useful. The file `sas-keyboardmacros.ahk` contains the most advanced topics.

CONCLUSION

SAS keyboard abbreviations are great, but they do have some disadvantages. AHK hotstrings offer similar functionality while removing most of the stumbling blocks associated with SAS keyboard abbreviations. If you like SAS keyboard abbreviations, then I encourage you to give AHK a try.

REFERENCES

AutoHotKey home page: <https://autohotkey.com/>

More details on hotstrings and hotkeys: <https://autohotkey.com/docs/Tutorial.htm#s2>

More details on escape characters: <https://autohotkey.com/docs/commands/EscapeChar.htm>

ACKNOWLEDGEMENTS

Thanks to Elizabeth Ceranowski for permission to use screen shots from her 2009 SGF paper on SAS keyboard abbreviations.

<http://support.sas.com/resources/papers/proceedings09/073-2009.pdf>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Shane Rosanbalm
Enterprise:	Rho, Inc
Address:	6330 Quadrangle Drive
City, State ZIP:	Chapel Hill, NC, 27517
E-mail:	shane_rosanbalm@rhoworld.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.