

Keyboard Macros - The most magical tool you may have never heard of - You will never program the same again (It's that amazing!)

Steven Black, Agility-Clinical Inc., Carlsbad, CA

ABSTRACT

Stop cutting and pasting old code into new programs, stop worrying about forgetting or misplacing really neat code, let SAS remember it for you! All through the use of Keyboard Macros! The term Keyboard macro is a mild misnomer as they are not really macros or at least not in the way that most SAS programmers think of macros. They should really be called Keyboard Rememberalls. Except they actually tell you what you forgot and they can hold all of your secret, and not so secret, code. The title may be an exaggeration but I have found these to really speed up my programming.

In this paper I will demonstrate how to create, use, and transfer keyboard macros. I will also discuss a few of the more in-depth aspects of Keyboard Macros with the intent that the reader will use and make them a part of their programming repertoire.

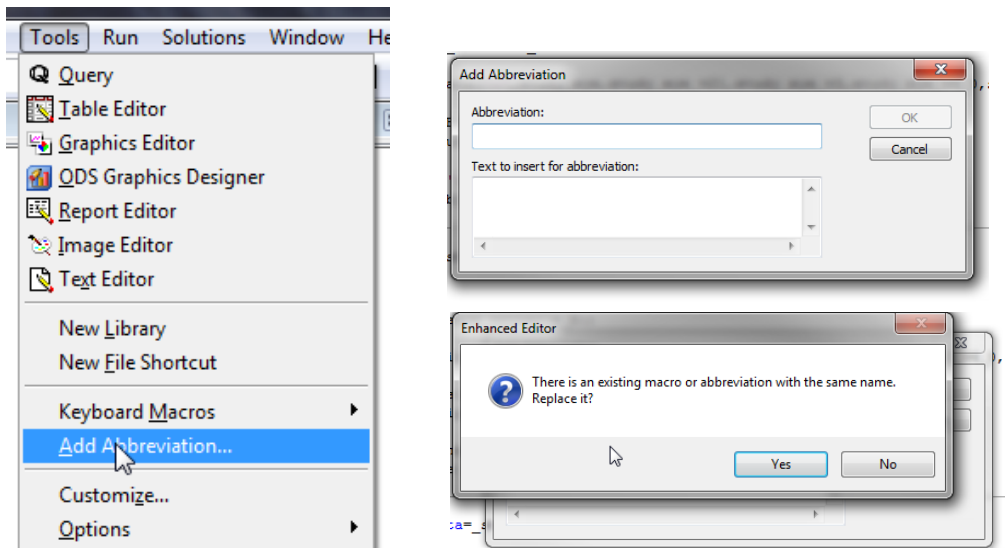
WHAT IS A KEYBAORD MACRO?

If you have ever searched through all of your programs looking for some useful code or find yourself copy/pasting large chunks of code from one program to another to save time but then end up having to tweak every little thing to get it to work. Keyboard macros are the solution you've been looking for.

A keyboard macro, also known as an abbreviation, basically retains selected code in SAS and will place this code into your enhanced editor window when the proper abbreviation is typed in. A common use is creating the program header required by various employers to state the program name, purpose, programmer, and change history. Once stored as a keyboard macro/abbreviation I can easily place this code on the top of every program then fill in the needed items.

CREATING A KEYBOARD MACRO

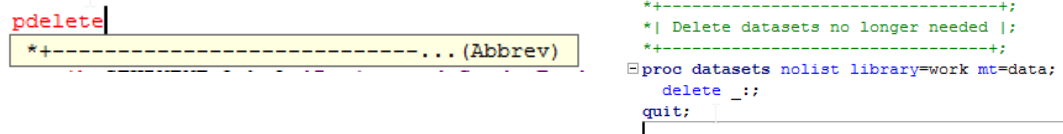
To create a keyboard macro/abbreviation simply select Tools > Add abbreviation or hold the **Shift + Ctrl + A** keys. It's best to copy the text that you'd like save first before bringing up the abbreviation window and paste it into the text portion of the box. Once the text has been placed in the box, create an abbreviation which will be used to pull up the text in the future. Use caution when creating the abbreviation name to not use the very common terms such as 'proc' or 'data' as every time you type this text in the enhanced editor the abbreviation process will be activated. If you name it the same name as another keyboard macro then you will receive a notice and can decided if you want to overwrite the previous abbreviation.



CALLING A KEYBOARD MACRO

To call or use the abbreviation simply type in the pre-specified abbreviation text specified into the enhanced editor window. Once the final letter is typed a small tan pop-up tip box containing the first few characters of the abbreviation will appear below the typed text. Pressing the TAB or ENTER key will place this stored text/code into the window.

The example below shows a quick deletion of the WORK library of datasets containing a '_' prefix.



The image shows a SAS code editor window. On the left, the text 'pdelete' is typed, and a yellow tooltip box appears below it containing the text '*+-----+ ... (Abbrev)'. On the right, the macro definition is shown:

```

*+-----+;
*| Delete datasets no longer needed |;
*+-----+;
proc datasets nolist library=work mt=data;
  delete _;
quit;

```

EFFECTIVE USES

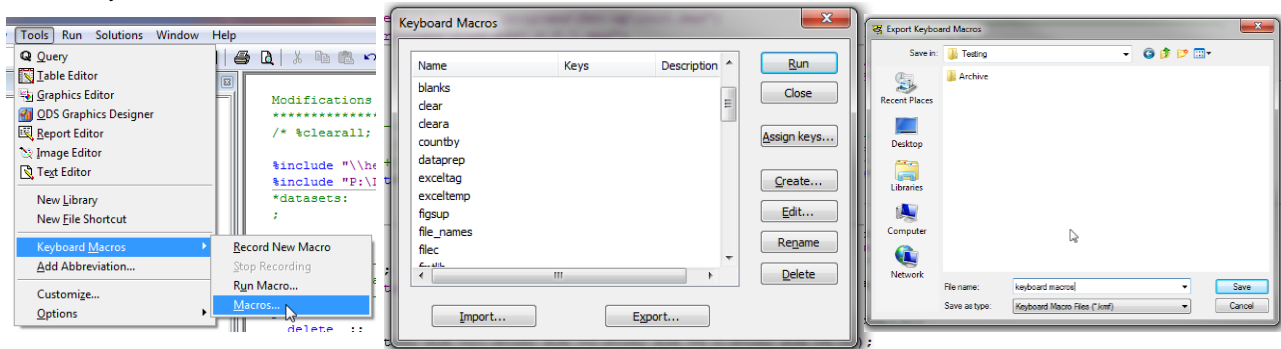
The term effective is quite subjective, however to further illustrate the usefulness of these I will provide a summary of how I have found them to be useful. I have mentioned previously the creation of program headers, and deleting data sets, other uses include:

Creating array templates, creating small blank line templates for missing frequency categories, clearing the log and output window, saving the code to count by a variable, preparing data for a PROC FREQUENCY, the common code used for excelxp tagsets, common PROC TEMPLATE code, common code for using the X command, the FMTLIB option in PROC FORMAT, greater than or less than symbol code for both unicode and font style, PROC COMPARE code, PROC FREQ code, PROC IMPORT code, PROC MEANS code, PROC UNIVARIATE code, PROC REPORT code, common PROC SQL code, PROC TRANSPOSE code, capturing the pvalue for both categorical and continuous data, rename syntax, retain example, company specific warning code, list of numbers 1-10, string of double quotes for easy format creation, inserting programmer's name and many more.

Instead of developing a large macro with 20 different parameters to try to create every table needed, I found that by creating common template codes I can simply paste in the code needed and then add a few variable names and I'm done.

MANAGING THE MACROS

Once these abbreviations have been created it's easy to forget abbreviations you used/assigned or what they do. You can easily see the entire list of abbreviations created by pressing **Ctrl + Shift + M**. Or by selection Tools>Keyboard Macros> Macros.



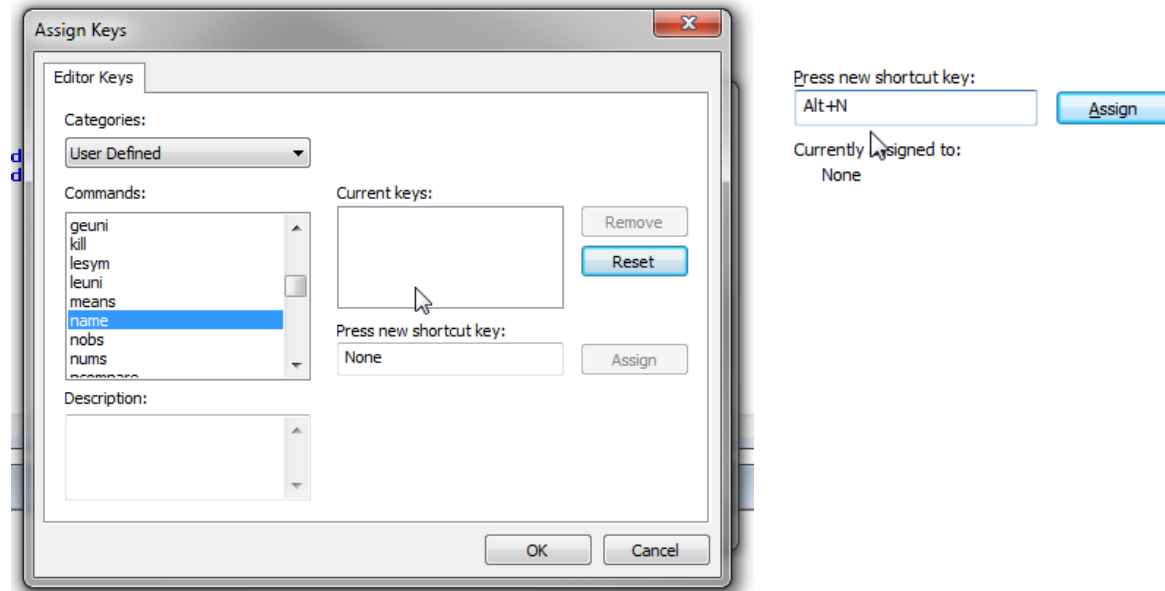
From this screen you can also edit, rename, and or delete previously created abbreviations.

Sometimes I forget what a keyboard macro does so I can either select edit or rerun it to see the code, or better yet, create a list of the abbreviations with a brief description and save this list as a keyboard macro for easy recall.

Switching between an office-based PC SAS to a server based VPN/ server based SAS has become more common and so it's very handy to be able to move these abbreviations between computers or share them with a co-worker. To import or export the macros, simply click on the appropriate button in the keyboard macros option box which will open up a file directory box to either retrieve or create a keyboard macro file (.kmf extension). When exporting you also have the ability to name the export file, select which abbreviations you will include and save the file in the desired location. When importing only .kmf files will be allowed to be selected.

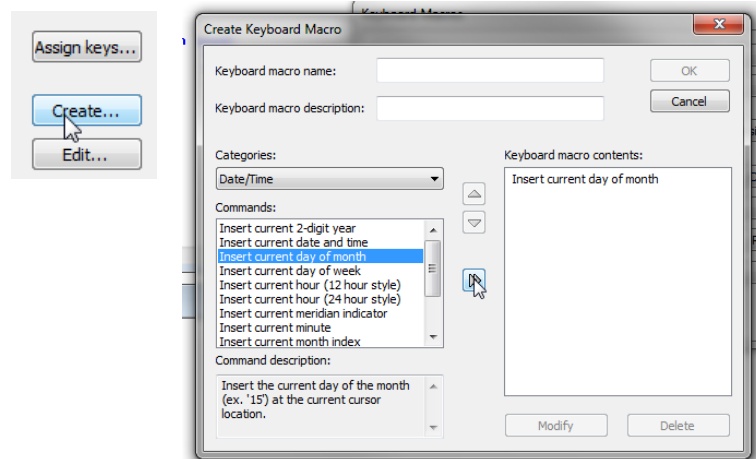
CREATING A SHORTCUT KEY

Within SAS there are a number of Shortcut keys that have been pre-defined. Some of these include changing the case of selected text (**Ctrl + Shift + U** or **Ctrl + Shift + L**), or commenting out text with line comments (**Ctrl + /**). Using the keyboard macro window you can easily create a keyboard shortcut for your own keyboard macros. If I wanted to create a shortcut for my name I would create an abbreviation for my name called name, then I would click on the 'Assign keys' button and select the abbreviation name. Then in the "Press new shortcut key:" box, I would assign the shortcut key by pressing my choice of shortcut keys such as **Alt + N**. By pressing the "Assign" button this would set-up the shortcut key. Now, whenever I want to type my name I can type ALT + N and it appears within the editor window.



CREATING THE CURRENT DATE KEYBOARD MACRO

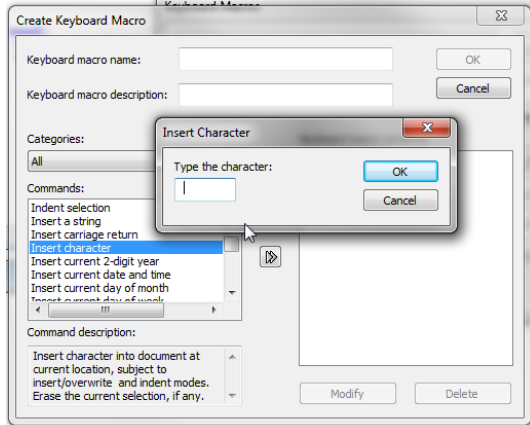
When creating a keyboard macro from scratch there are a number of available options that have not been assigned a shortcut key. These can be found in the commands box. In particular the options in Date/Time category can be quite helpful. In order to use these options we simply need to assign a shortcut key and/or abbreviation. In most programs when I enter the date I like to do so using a date9. format of DDMMYYYY. However there is not a date option that gives me the date in the format that I want so using the following steps I can create my own abbreviation with the current date.



In the main keyboard macro box (**Ctrl + Shift + M**) I select 'Create...'. This opens up a new box containing all of the possible keyboard macro options. I then search for the 'Insert current day of the month' command and move it into the selected box by using the double arrow button. I then repeat these steps for inserting the current month name

and insert current year. If I leave it as and fill in the keyboard macro name and provide a description then I would have nice abbreviation with today's date. However, since I want true date-9 format I need to add a few more steps.

First I'll need to create a quick way to find the start of the month once created. I do this by inserting a blank space between the day and month by using the 'Insert character' option which, when added, allows me to choose the character wanted, I use a space. I then need to upcase the first three characters of the month and remove any extra month characters.



This I can do by adding and organizing the commands in this order, using the up and down arrows.

1. Insert current day of the month
2. Insert character (' ')
3. Insert current month name
4. Move cursor to previous word start
5. Delete previous character
6. Convert selected text to uppercase
7. Convert selected text to uppercase
8. Convert selected text to uppercase
9. Delete to next word start
10. Insert current year

Once these steps are complete and a name has been assigned click 'OK'. Then select 'Assign Keys' and assign it a key as illustrated previously. I use the ALT keys for my homegrown shortcuts as these generally have no other prior designation so the full alphabet is available without fear of overwriting a previous assignment.

There are many commands available which SAS® programmers may not be aware of, but it is hoped that once these options are reviewed that additional macros will be created to assist in the creation of more tools to enhance efficiency.

CONCLUSION

In this paper I have illustrated the amazing capability of keyboard macros and have provided a list of potential uses. I have also provided additional insight into the some of the more complicated steps that can be used to quickly insert the programmer's name and current date into their programs. I hope that this paper provides the framework for further use and development of your own keyboard macros.

REFERENCES

Carpenter, Arthur L. 2012 Doing More with the SAS® Display Manager: From Editor to ViewTable - Options and Tools You Should Know" *Proceedings of the SAS Global 2012 Conference*. Cary, NC: SAS Institute Inc.
<http://support.sas.com/resources/papers/proceedings12/151-2012.pdf>

Houston Baptist University: SAS Enhanced Editor Tips (Author unknown)
https://hbu.edu/HBU/media/HBU/publications/presentations/SAS_Enhanced_Editor_Tips.pdf

CONTACT INFORMATION

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

Name: Steve Black
Enterprise: Agility-Clinical Inc.
Address: 6005 Hidden Valley Rd.
City, State ZIP: Carlsbad CA, 92011
Work Phone: 760-658-5919
E-mail: steven.c.black@gmail.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.