

Abbreviation Creation and Use: Loading an Abbreviation file

M. Chris Moser, Contractor for DataMeans, USA

Ed Heaton, Westat, USA

Russ Lavery, contractor for Numeric Resources, USA

ABSTRACT

This paper focuses on importing the SAS ® “abbreviation file” that is downloadable from the DataMeans web site. The abbreviation file adds context sensitive help to your SAS Enhanced Editor and contain abbreviations, syntax, options and hints for coding Base SAS procedures and functions (as well as some samples of hashing, binary searches and compressed index use). These abbreviations are simpler and less complete than SASHELP but might be useful to new programmers. Papers in the reference section discuss details of abbreviation creation and management for those who wish to create their own abbreviations or to customize the files created by the authors.

INTRODUCTION

Many programming languages have context sensitive help and this feature is especially useful when a programmer *almost* knows the syntax of the function or procedure s/he is wants to type into code. SAS has had context sensitive help for many years, but the user has had to create their own help via an editor function called SAS Enhanced Editor Abbreviations. What is not commonly known is that these abbreviations, once created, can be exported to a file and then imported into SAS on another machine. The authors have created abbreviation files for most of the SAS v9 functions and for many common base SAS procedures. The convenience of not having to create abbreviations is the major deliverable of this paper. Hopefully, the inclusion of hashing, binary searches and compressed indexes will make those techniques a bit easier to learn.

A few years ago, Ed Heaton created, exported and put into the public domain a file of SAS function abbreviations. This work motivated Moser and Lavery to create a file of abbreviations for many of the procedures in Base SAS. This paper outlines how to import this file of abbreviations so that your SAS system can give you context sensitive help. It is expected that the file will be updated as time goes by. The most current version of the file can be downloaded from the DataMeans web site www.datameans.com.

ABBREVIATIONS - HOW TO USE THEM

To the right we see how to use abbreviations.

When the programmer types a string previously defined as a “reserved abbreviation string” (here “proc freq” is the string) the editor displays a note (that ends in “Abbrev”) in a box.

If the user hits either the tab or Enter key, the abbreviation replaces the “reserved abbreviation string”. If the user types any other key, the box goes away. It is that simple.

Abbreviation recognition is case sensitive.

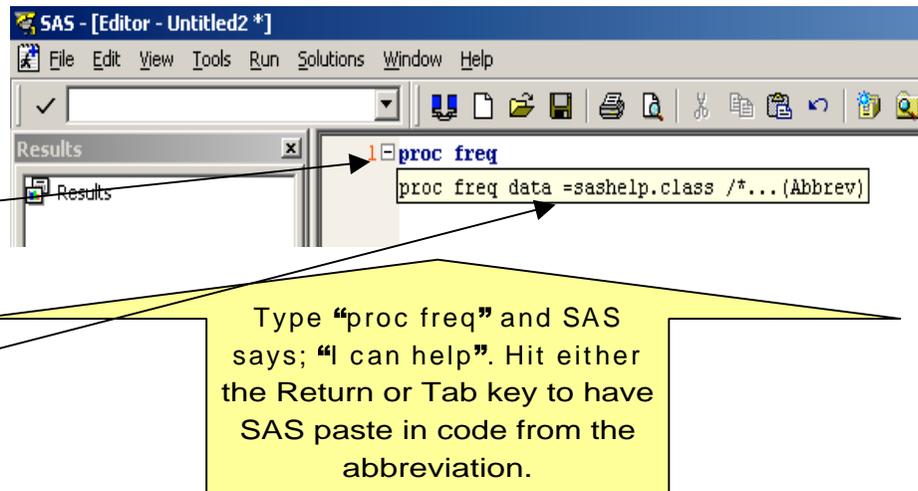


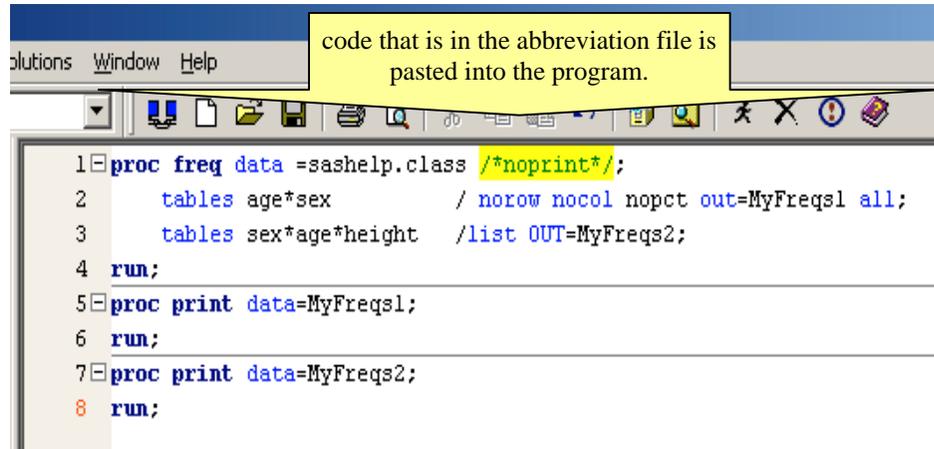
FIGURE 1

PhUSE 2008

We see, to the right, that the string “proc freq” was replaced with an example of the proc from the abbreviation file that the authors are putting in the public domain.

Abbreviations for procs contain examples that run.

It is expected that some programmers will want less in their abbreviations than is in the files that the authors provide. In this case, the programmer can easily modify the abbreviation.



```
1 proc freq data =sashelp.class /*noprint*/;  
2     tables age*sex          / norow nocol nopct out=MyFreqs1 all;  
3     tables sex*age*height   /list OUT=MyFreqs2;  
4 run;  
5 proc print data=MyFreqs1;  
6 run;  
7 proc print data=MyFreqs2;  
8 run;
```

FIGURE 2

The “reserved abbreviation strings” in the file provided by the authors were mostly created in lower case. This means that a programmer who imports these abbreviations can avoid having the boxes constantly appearing by simply using mixed case when typing. Because the string recognition is case sensitive, typing “proc freq” will cause SAS to display a box and an offer to paste in the abbreviation. Typing “Proc Freq” or “PROC FREQ” will not cause SAS to display a box. Additionally, if the programmer does not hit a tab, or enter, key, the box goes away.

If you mistype a string (for example, “proc freeq”) and *want* the abbreviation, all you must do is delete the misspelling – starting from the rightmost character-and then correct the misspelling. In the example of proc freeq above, you would hit backspace until the editor displays “proc fre” and then type a “q” to have the SAS Editor display the box that is shown in figure 1.

ABBREVIATIONS - HOW TO IMPORT THE ABBREVIATION FILES

Data Means is hosting a file of abbreviations that contains abbreviations for SAS functions that were put into the public domain by Ed Heaton from Westat, Inc. and added to by Moser and Lavery.

While not a replacement for the SASHelp files, the abbreviations for procedures contains detailed information about a procedure (usually complete code snippets as can be seen in figure 1).

Importing the abbreviations is very easy. Download the file from www.DataMeans.com and store it in an accessible place. Then start SAS and activate the Enhanced Editor.

From the Enhanced Editor menu, select tools-> keyboard macros -> macros. This brings up the Keyboard Macros dialog box shown to the right. Click on Import, navigate to your file, select the file and click OK. That is all there is to it.

Keyboard Macro dialog box:

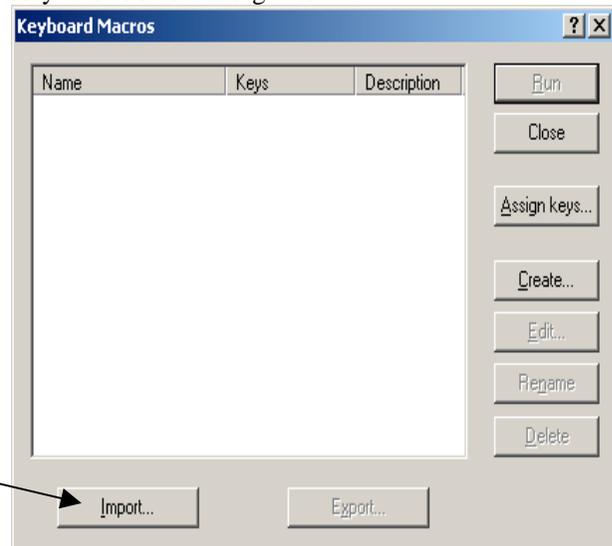


FIGURE 3

It should be mentioned that there is a limit on the number of abbreviations that can be loaded into SAS and that importing the file created by the authors will not allow any more abbreviations to be created without removing an abbreviation. To fit in abbreviations that the authors thought to be interesting, the Min and Max functions were removed from the KMF file of abbreviations. This allowed the creation/importing of some abbreviations (Proc Report and Proc Tabulate) that the authors judged more interesting and more difficult to remember than Min and Max. SAS is discussing, but not committed to, expanding the abbreviation capacity after work on 9.2 is completed.

PhUSE 2008

If your Keyboard Macros dialog box looks like Figure 4, click close and start using the abbreviations.

Abbreviations can be managed (created, deleted, edited, linked to keys, exported, etc.) using the Keyboard Macros dialog box. Interested users can read the papers in the reference section to learn these skills.

Examples of big file techniques (hashing, binary searches and compressed indexes) were mentioned before in this paper (Thanks to Dorfman and Keintz for putting them in the SAS literature). It is unlikely that a programmer would type, in a program, the exact string that triggers these abbreviations. These abbreviations are best accessed by opening the keyboard macro dialog box (right) selecting an abbreviation with an interesting name and clicking the "Run" button.

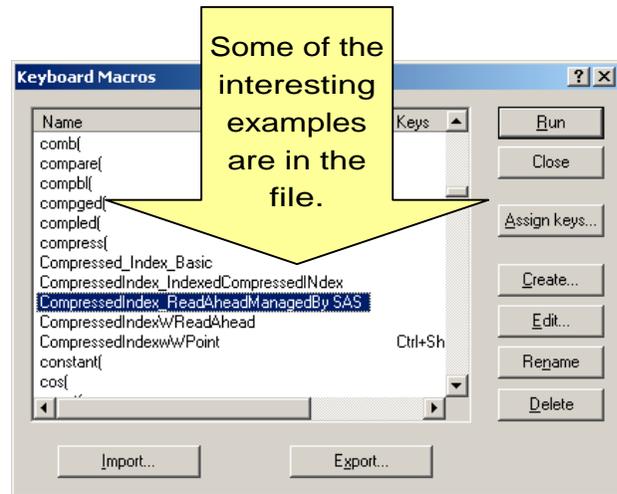


FIGURE 4

SUMMARY

This paper discusses a file of abbreviations that can be downloaded from the DataMeans web site. This file can be easily imported into your system, giving you context sensitive help without having to create abbreviations yourself. The procedure abbreviations contain examples using options that might not be familiar to new programmers. It is hoped that these procedure abbreviations will help new programmers learn SAS quickly. The hashing, binary searches and compressed index examples that are in the abbrev file might be fun to check out.

The authors had more things that they wanted in the abbrev file than could be loaded into SAS. They removed the abbreviations for the max and min functions so that more interesting (and harder to remember) abbreviations could be added. Download the file from www.DataMeans.com and

References

Ceranowski, Elizabeth. "SAS Abbreviations are your friends, use the template method to code!" *Proceedings of the 2008 PharmaSUG Conference Paper SA07*
<http://www.lexjansen.com/pharmasug/2008/sas/sa07.pdf>

Carpenter, Arthur L. "Creating Display Manager Abbreviations and Keyboard Macros for the Enhanced Editor" *Proceedings of the SUGI 28 conference paper 108-28*
<http://www2.sas.com/proceedings/sugi28/108-28.pdf>

Grant, Paul. "Creating Code Templates in the SAS Enhanced Editor Using Abbreviations and User Defined Keywords" *Bay Area SUG Quarter1, 2006*
http://www.basug.org/downloads/2006q1/code_templates.pdf

DiTommaso, Dante, Szilagyi, Benjamin. "Lessons Learned: Valuable (But Hidden) SAS Details" *Proceedings of the 2006 PhUse Conference Paper cs03" PhUSE 2006*
<http://www.lexjansen.com/phuse/2006/cs/cs03.pdf>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Ed Heaton Senior Systems Analyst, Westat Inc., Rockville MD, USA Email EdHeaton@WESTAT.COM

M. Chris Moser,
Senior Consultant for DataMeans Corp.
Colonia, NJ, USA
Email: Cmoser@DataMeans.com

Russell Lavery,
Contractor for Numeric Resources
Ardmore, PA, USA
Email: russ.lavery@verizon.net

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.