

# Managing the Organization of SAS® Format and Macro Code Libraries in Complex Environments Including PC SAS, SAS® Enterprise Guide®, and UNIX SAS

Roger D. Muller Ph.D., Data-To-Events, Inc., Carmel, IN

## ABSTRACT

The capabilities of SAS® have been extended by the use of macros and custom formats. SAS macro code libraries and custom format libraries can be stored in various locations, some of which may or may not always be easily and efficiently accessed from other operating environments. Code can be in various states of development ranging from global organization-wide approved libraries to very elementary “just-getting-started” code. Formalized yet flexible file structures for storing code are needed. SAS user environments range from standalone systems such as PC SAS or SAS on a server/mainframe to much more complex installations using multiple platforms. Strictest attention must be paid to (1) file location for macros and formats and (2) management of the lack of cross-platform portability of formats. Macros are relatively easy to run from their native locations. This paper covers methods of doing this with emphasis on: (a) the option “sasautos” to define the location and the search order for identifying macros being called, and (b) even more importantly—the little-known SAS option MAUTOLOCDISPLAY to identify the location of the macro actually called in the saslog. Format libraries are more difficult to manage and cannot be created and run in a different operating system than that in which they were created. This paper discusses the export, copying and importing of format libraries to provide cross-platform capability. A SAS macro used to identify the source of a format being used is presented.

## INTRODUCTION

Beginning SAS programmers tend to pay very little attention to the long-term usage and storage of custom formats and user-coded macros. Often the programmer just includes the code in their single program file. Then as needs grow, they place the code in multiple files. Then all of a sudden their needs expand and they realize the need for some sort of folder (directory) structure to aid in keeping things straight. Then the scope of their activity really increases when multiple users are involved with a project. Then mass confusion and frustration set in when the programmers inherit code from others and must use and maintain it. Commonly heard: “Where did this get called in from?” This paper has goals to address the following in a simple straightforward discussion stressing the similarities between maintaining custom formats and macros, not the differences.

1. Folder (directory) structures. Keep them simple and maintainable. If you do not, the usage will fall apart, even if you the creator are the only user.
2. How to control the search order for the directories that SAS searches to find a macro or a format. This puts the programmer in full control.
3. Two simple tools to print back to the log exactly where SAS found the code. For macros, this is a little know SAS option; while for formats it is a downloadable macro from support.sas.com.

A 15-minute video of this presentation is available on Youtube.

## LIBRARY STRUCTURES

Keep the library structures very simple. Try to follow these guidelines:

1. Folder (directory) structures. Keep them simple and maintainable. The libraries should have only the level of structural detail that you are willing to maintain. If you do not, the usage will fall apart regardless if this structure is only for your use or for the enterprise.
2. Folder (directory) naming as well as file naming: use meaningful names and use no capital letters or spaces. Why? These work fine in Microsoft Windows environments but are horrible in Unix. Spaces are not permitted in Unix. Therefore use underscores. Upper case and lower case letters are interchangeable in Windows, they are not interchangeable in Unix. A file or folder with one capital letter in its name is totally different from what appears to be a file of the same file without the capital in Unix. Make your code transportable.

3. Maintenance – carefully consider who has write access, who has read access, who is in charge of updates, how will upgrades be announced, how will older versions of the code be maintained if older programs have to be rerun at a later date.
4. Don't overdo it. Libnames have to be established for every directory, or the exact file path has to be specified in the search path with certain options and statements. Fewer directories may be better than many.
5. Avoid nesting of folders within folders. Things can get very complicated to unravel if problems are encountered
6. If macros and formats are shared, make sure you routinely communicate changes, use directions, impact on previously run programs, etc. to the user base.

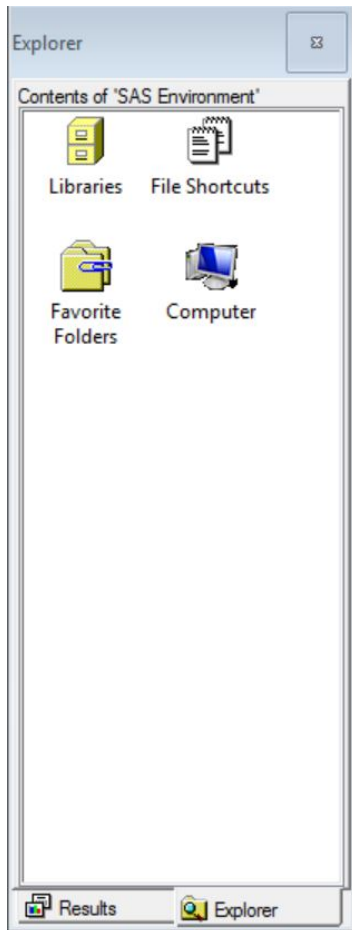
### *Library Structures For Formats*

Library structures for formats require a little more discussion than those for macros. Part of the difficulty in understanding how formats are stored stems from the initial training examples in assorted SAS documentation. Remember that in all cases, custom formats are produced using Proc Format. There is both a “source code” file and a compiled file that is actually a SAS catalog.

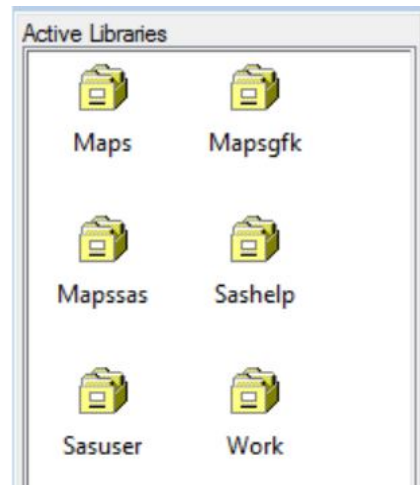
Here is a typical example of simple format coding..

```
*Use of format with no library specified (placed in a work catalog);
proc format;value speedw 100='Slow' 150='Medium' 200='Fast';run;
```

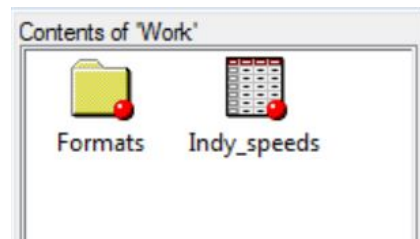
What happened? A format called speedw was created. But where is it? Using the SAS Explorer in interactive PC SAS, follow the steps in Display 1 thru Display 5. Not only will you find it, but you can see the values in it (but can't edit it via this process).



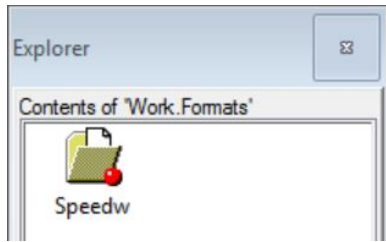
**Display 1. Launch SAS Explorer**



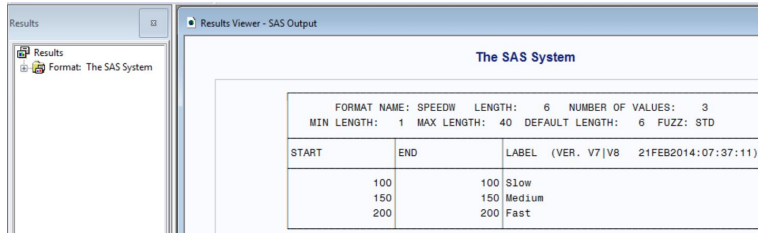
**Display 2. Select Work Library**



**Display 3. Contents of Work – Open the SAS Format Catalog**



**Display 4. Open Format Speedw**



**Display 5. View of Format Speedw**

### *More Complex Format Library Structures*

Now let's look at a much more complex example. Formats are going to be written off to 3 different libraries with 2 different member names in each. Some of the formats will be written with "one-level" names and others with "two-level" names. (Note: do not confuse this with one- and two-level SAS datasets names. While there is some similarity in concept, these are catalog entries, not datasets). To make this situation potentially more complex, the same format name is going to be repeated in a few places. This could be viewed as the beginnings of a true mess, yet it is not unlike many situations encountered on a day-to-day basis, particularly if working on preexisting code in a team environment.

\*\*\*\*\*

\*Assortment of Format Libraries -- A potential for a mixed up mess if not well understood;

\*Note: Format speed. appears in multiple places;

\*PERSONAL FORMAT LIBRARY;

libname persfmt 'C:\Users\RogerMuller\Documents\My SAS Files\Formats\Formats\_Personal\';

proc format library=persfmt; \*1 level name for catalog;

value speed 100='Slow' 150='Medium' 200='Fast';run;

proc format library=persfmt.rate\_speed; \*2 level name for catalog;

value speed 100='Slow' 150='Medium' 200='Fast';run;

\*DEPARTMENTAL FORMAT LIBRARY;

libname deptfmt 'C:\Users\RogerMuller\Documents\My SAS Files\Formats\Formats\_Departmental\';

proc format library=deptfmt; \*1 level name for catalog;

value speedd 100='Slow' 150='Medium' 200='Fast';run;

proc format library=deptfmt.rate\_speed; \*2 level name for catalog;

value speedrs 100='Slow speedz' 150='Medium speedz' 200='Fast speedz';run;

\*CORPORATE FORMAT LIBRARY;

libname corpfmt 'C:\Users\RogerMuller\Documents\My SAS Files\Formats\Formats\_Corporate\';

proc format library=corpfmt; \*1 level name for catalog;

value speed 100='Slow' 150='Medium' 200='Fast';run;

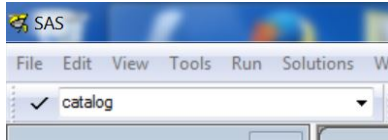
proc format library=corpfmt.rate\_speed; \*2 level name for catalog;

value speedc 100='Slow' 150='Medium' 200='Fast';run;

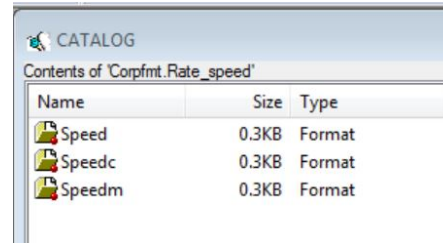
\*\*\*\*\*

## What has happened?

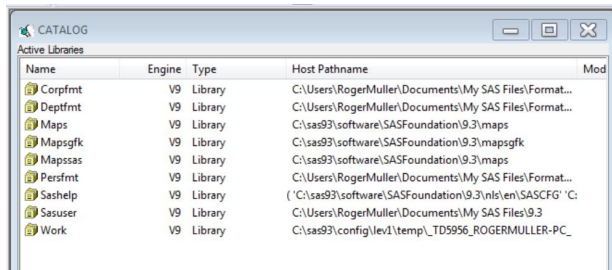
Note that the folders (directories) were all made before the code was run. What has happened to the assorted formats as they have been placed in SAS catalogs? Use the command “Catalog” in the PC SAS command box (Display 6 ). The functionality to look at catalogs is provided by an “Add-In-Tool” for SAS Enterprise Guide that is available via download. (See Hemedinger publications).



Display 6. Entering Catalog on Command Line



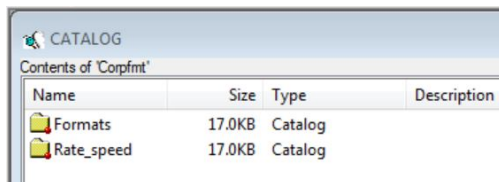
Display 9. Select Speedc Format



Display 7. Resulting Catalog Display, Open Corpfmt

START	END	LABEL	(VER. V7 V8	21FEB2014:08:10:51)
100	100	Slow		
150	150	Medium		
200	200	Fast		

Display 10. Display Format Speedc



Display 8. Select Rate\_speed Member Name

Notice the assorted one level and two level names. Furthermore you can look at the actual values in the format (Display 10).

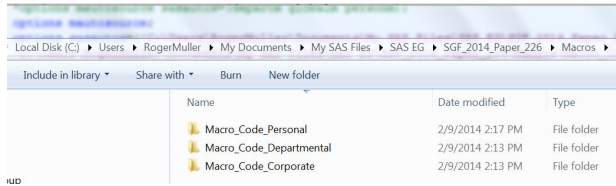
**Important:** While the coding shown above makes it appear as if the location of the SAS format libraries is very flexible (and indeed it is), there is one major restriction. The format catalogs must have been compiled on the same operating system as the operating system on which they are being called by SAS. In other words, while you could link to a folder containing SAS formats on Unix for a Windows SAS session, you can't use those formats in Windows PC SAS. To move formats across operating systems, one option is to copy the underlying SAS code that developed the formats from the source directory over to the current machine and rerun. The other alternative is to use a transport option within Proc Format to export and import the formats (cntlout, cntlin options).

## Library Structures for Macros

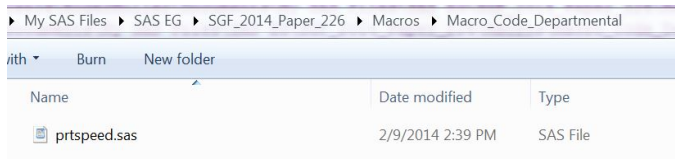
Library structures for macros are much simpler to understand do to the fact that all you are storing is SAS code. There is a little used macro compilation capability that will not discussed in this paper. Likewise, the use of Stored Processes on newer SAS systems implementing Integration Technologies is not addressed. Macros can be stored on any drive on any operating system to which you have access (which in some cases may be read only).

The structure of the directories/folders may be as complicated or as simple as you wish. Hierarchical structures may be actual or logical. In the following figure, there is a logical hierarchy from corporate to personal, yet all of the folders are at the same level on the drive (Display 11). The macro file in the folder and the actual macro code are

displayed in Displays 12 and 13.



**Display 11. Proposed Folder Structure for Macros**



**Display 12. The Macro File Inside Folder**

```
%macro prtsspeed;
proc print data=indy_speeds;
title "Dataset Speed, I was printed by departmental macro";
run;
%mend prtsspeed;
```

**Display 13. The Actual SAS Code in the Macro File**

## CONTROLLING THE SEARCH ORDER

The SAS programmer has complete control over the search orders for finding the first location of a format or a macro. The process is quite simple and very similar for each.

### Controlling Search Order for Formats

To control the search order of format libraries for SAS to find the FIRST OCCURRENCE of the format, use options ffmtsearch.

```
*****
options ffmtsearch=(work persfmt persfmt.rate_speed deptfmt deptfmt.rate_speed corpfmt corpfmt.rate_speed);
*****
```

By changing the order of these library/catalog one or two level names, you are in control of the search order. Note how the two-level names are specified.

### Controlling Search Order for Macros

If your macros are in folders, you control the search order by the use of the sasautos= option. It is important to remember to turn on the option maautosource to enable SAS to utilize the sasautos functionality. Nearly all large SAS systems are built with macros that are “autocalled” out of their respective libraries. The SAS programs that contain the macro calls are frequently called “driver files”. The alternative to using this methodology is the older %include statement. As with formats, the macro at the first location in the search path that contains the macro will be the one that is utilized. Note that reference is made to sasautos which is the location of the 16 macros furnished with the base SAS installation.

```
*****
options sasautos=('C:\Users\RogerMuller\Documents\My SAS Files\SAS
EG\SGF_2014_Paper_226\Macros\Macro_Code_Corporate' 'C:\Users\RogerMuller\Documents\My SAS Files\SAS
EG\SGF_2014_Paper_226\Macros\Macro_Code_Departmental' 'C:\Users\RogerMuller\Documents\My SAS
Files\SAS EG\SGF_2014_Paper_226\Macros\Macro_Code_Personal' 'sasautos);
*****
```

Yes this is tedious. But, once established is very easy to maintain.

**Important:** SAS macros can be kept on another operating system (e.g. Unix) and called into the current operating system (e.g. PC SAS) and utilized with no modification other than to some statements contained within the code that might be operating system specific (such as libnames).

## CONFIRMING THE LOCATION OF THE MACRO OR FORMAT THAT WAS ACTUALLY USED

The potential exists for the macro or format that SAS actually found and called is different from the one that YOU THINK it called. Typical symptoms of this are: "...I changed the code and I know it is correct, yet it looks like SAS didn't event read it. What's going on?...".. Odds are that what you changed is not in the location that SAS called. How can you confirm what is actually happening?

### *Confirming the Location of a Format SAS Actually Used*

There is a reasonably simple way of doing this. There is a macro at [www.support.sas.com](http://www.support.sas.com) named %findfmt that will spell out the location on the SAS log. This macro (see references for location) has been modified by the author following download to also report the member name in cases where a 2-level catalog naming convention is being followed.

To use the macro call it with arguments of the format name and the letter "f" for formats and "i" for informats.

```
*****  
%findfmt(speedc,f);run  
*****
```

The following is returned to the saslog

```
Format SPEEDC is user-defined and exists in the CORPFMT library RATE_SPEED memname  
NOTE: There were 1 observations read from the data set WORK.TEST.
```

#### Display 14. Information That Macro %findfmt Returns to SAS Log

The first occurrence of the speedc format is in the corpfmt library in memname rate\_speed. PRETTY SLICK!

### *Confirming the Location of a Macro SAS Actually Used*

There is a little-known option in SAS that will tell you exactly what the location of the sasuto macro was. IF THE SASAUTO CAPABILITY IS BEING USED, MAUTOLOCDISPLAY SHOULD ALWAYS BE TURNED ON. There is no reason not to have it turned on as the information returned to the log is compact, extremely informative and easy to read.

```
*****  
Options mautoocdisplay;  
*****
```

After calling a custom user-defined macro named prtsped, here is the information that option mautoocdisplay placed on the log. PRETTY SLICK!

```

127 %prtspeed;      *this is the macro being called by the sasautos
127! sort order;
MAUTOLOCDISPLAY(PRTSPEED):  This macro was compiled from the autocall
                             file C:\Users\RogerMuller\Documents\My
                             SAS Files\SAS
                             EG\SGF_2014_Paper_226\Macros\Macro_Code_Co
                             rporate\prtspeed.sas

NOTE: There were 3 observations read from the data set
      WORK.INDY_SPEEDS.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.03 seconds
      cpu time            0.00 seconds

```

## Display 15. Information Returned to Log by Option Mautlocdisplay

## THE BIG QUESTION: MAINTAINING THE LIBRARIES??

Format and library maintenance requires some very careful consideration of a number of issues. Maintenance, version control and archiving being three of extremely high importance for any user group ranging in size from one to the entire enterprise. This essentially becomes a technical management/communication issue.

### What about archiving off a project?

When the time comes to archive a project that must have retrievable code for either foreseen or unforeseen reasons, how should the “current” versions of assorted macros and formats that are out in libraries be retained. There probably is not “one size fits all” solution. About the only absolutely correct answer with minimal potential failure is to copy them to an archive directory that also contains the sascode for the project. Macros once identified are easy to copy over. Formats are somewhat more challenging. If you have the source code, you may wish to copy that and rerun it if needed. Or, you may want to use the cntlout and cntlin options on proc format to copy formats out and into an archive.

## CONCLUSION

This paper has addressed the issues of library file structuring, controlling search order and confirming usage for both SAS formats and SAS macros in a very straight-forward simple manner. A much more complex presentation and paper has been accepted for presentation at the 2014 Pharmaceutical SAS Users Group meeting. The paper will expand on issues covered here. In addition, there is a 15-minute Youtube video of this presentation.

## REFERENCES

- Arthur L. Carpenter, “Building and Using Macro Libraries”, SUGI27 Proceedings, <http://www2.sas.com/proceedings/sugi27/p017-27.pdf>
- Ronald Fehd, “A SASAutos Companion: Reusing Macros”, SUGI30 Proceedings, <http://www2.sas.com/proceedings/sugi30/267-30.pdf>
- Harry Droogendyk, “Which SASAUTOS Macros Are Available to My SAS® Session?”, SUGI 2009 Proceedings [support.sas.com/resources/papers/proceedings09/076-2009.pdf](http://support.sas.com/resources/papers/proceedings09/076-2009.pdf)
- Macro findfmt, “Usage Note 24019: Is there a way to determine whether a format or informat is available for use, and whether it is supplied by SAS or user-defined?”, <http://support.sas.com/kb/24/019.html>
- Chris Hemedinger, “Using Custom Tasks in SAS Enterprise Guide”, <http://www.sas.com/offices/NA/canada/downloads/UserGroups/TASS-Fall2012/Hemedinger-CustomTasksinEG.pdf>
- Chris Hemedinger, “Viewing SAS catalogs from SAS Enterprise Guide”, <http://blogs.sas.com/content/sasdummy/2010/05/10/viewing-sas-catalogs-from-sas-enterprise-guide/>
- Chris Hemedinger, “Special steps for installing the custom tasks that you download”, <http://blogs.sas.com/content/sasdummy/2010/05/10/viewing-sas-catalogs-from-sas-enterprise-guide/>
- Roger Muller, 2014, “Managing the Organization of SAS® Format and Macro Code Libraries in Complex Environments Including PC SAS, SAS® Enterprise Guide®, and UNIX SAS”, 2014 SAS Global Forum Proceedings
- Roger Muller, “You Tube SAS Videos Channel”,

<https://www.youtube.com/playlist?list=PLdnXxLhLh0MzHry7j9GrSmFbBiN-Firgc>

## **ACKNOWLEDGMENTS**

The author acknowledges the assistance of Mr. D.J. Penix, President Pinnacle Solutions for help and advise on many SAS projects over the years.

## **CONTACT INFORMATION YOUR COMMENTS AND QUESTIONS ARE VALUED AND ENCOURAGED. CONTACT THE AUTHOR AT:**

Name: Roger D. Muller, Ph.D.  
Organization: Data-To-Events, Inc  
Address: 14475 Stephanie St.  
City, State ZIP: Carmel, IN 46033  
Work Phone: 317/985-0132  
Email: [rdmuller@hotmail.com](mailto:rdmuller@hotmail.com)  
Web: [www.data-to-events.com](http://www.data-to-events.com) [www.rogermullervideography.com](http://www.rogermullervideography.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.



