Paper 188-2012

What Is a SAS® Mentor and Why Do I Need one?

Stanley Fogleman, Harvard Clinical Research Institute, Boston, MA

ABSTRACT

A SAS mentor can be a valuable resource to junior programmers just starting out, or programmers who may have used other high level languages and are just starting to use SAS. The paper will focus on preparing a plan for a hypothetical junior programmer, with the goal of becoming a more proficient programmer over a one or two year period. SAS, unlike some other languages does not lend itself to being self-taught and some things that might look like an obvious choice can actually lead to performance penalties. The most important thing that mentors can provide is guidance – for things worth studying more and things that might be interesting, but you might only use infrequently.

INTRODUCTION

I learned SAS in a very haphazard fashion. My original goal was to learn as much about SAS as I could within a six month timeframe, because this was the length of the contractor's commitment to that company. I took a SAS Programming I course fairly early on, but I think it was five or six years before I took my first macro course! I had little or no guidance in choice of courses other than immediate project needs. As a side note, I took several years of C and C++ courses, and other than learning why you can call a function inside a function, very little was applicable to my needs as a SAS programmer. A beginning programmer faces a bewildering number of possible training paths. Their task in learning is probably in competition with the need to have them become productive quickly in their current position. Of course, position requirements change over time, and so, making someone productive in their current role works against someone's ability to be productive in other roles over time. Perhaps you have worked with people who have spent vast amounts of time solving problems that didn't need solving. (Parsing an RTF file and reading an XML file as text are two examples that come to mind.) If SAS procedures are a "black box" and the person using them is not sure how they work, but has some code that worked a few weeks ago, watch out!

PREREQUISITIES

Executive acceptance is certainly a must as a significant amount of time must be devoted to the care and feeding of a junior SAS programmer. Also, the changing of a corporate culture can present some unexpected roadblocks. There is a tendency at the executive level to think that all employees, even junior ones, arrive well trained. A pilot program with one or two junior programmers would be a great way to start. Presenting a prospective trainee to an executive with a completed plan (see next paragraph) should increase the chances of success.

MAKE A PLAN!

It is probably most realistic to plan for a one to two year time frame. For a hypothetical programmer just out of college, one might suggest the following courses:

- SAS Programming 1 and 2.
- SAS Macro Language 1 and 2
- SAS Enterprise Guide I Querying and Reporting

Make plans to attend a local, regional or national SAS conference. Often the regional and national conferences have learning tracks especially for the beginner. I often make the point to management that a three-day conference can be the equivalent of a course and more, since one gets a chance to interact with other SAS users. Also, the demo areas typically have a "code clinic" where one can bring problem code.

If developing a form would provide a context for this training plan and formalize it, then by all means develop one. Some other possible benefits would be the ability to develop a training budget and provide some guideposts (or milestones, if you prefer) for measuring progress.

WHAT SHOULD A SAS MENTOR DO?

The definition could vary widely from company to company, and even department to department, but above all, provide guidance. This could come in several forms:

1. Preparing a learning plan for a junior programmer.
2. Providing coding guidelines.
3. Helping to prepare an internal web page with links to SAS Training, SAS Support and SAS Documentation.
4. Being available for questions and concerns.
5. Providing guided learning.
6. Providing training milestones and keeping track of accomplishments.
7. Being a sounding board and devil's advocate for programming ideas and strategies.

WHAT SHOULD A SAS MENTOR NEVER DO?

- Become a substitute manager and/or supervisor
- Become a micromanager
- Be a substitute for formal training
- Be a "Monday morning quarterback."

PROVIDE CODING GUIDELINES

A good set of coding guidelines can make the difference between a well-structured environment and coding anarchy. Some pointers follow.

- Always include a header paragraph stating (at a minimum) the program name purpose and directory
- "One semicolon per line" can do wonders for readability.
- Make an effort to use self-documented variable names and programs whenever possible. Col1Row3 may describe a screen location perfectly, but it is of little value in determining what the variable is used for (or what it should contain!)
- Build a set of templates for frequently used routines.
- Comment early and often. Take pity on the person who has to maintain your code, especially at 2 in the morning, or when you are under pressure to meet a deadline and some particular patch of code isn't working right. Also, six months later when you are asked to make modifications, the simplest code can be mystifying without some explanation.
- Use SAS Data Set Labels – often one has little choice in choosing a name for a SAS Data Set to be given to an end user, but one has approximately 200 characters to describe the contents of a sas dataset in a sas label.
- Indentation is a great way to improve program clarity along with its cousin whitespace. Some people confuse compact code with "impacted code".
- Avoid implicit type conversions. The PUT and INPUT functions are ALWAYS preferable to dividing a character variable by 1 to "turn it into" a number.
- Data Steps should be succinct. Data Steps that span several paragraphs (and pages) are difficult to comprehend, let alone modify and/or maintain.
- At the earliest possible date, incorporate SQL code into your programs. Sometimes the discipline required to choose only the columns that are pertinent as well as their selection criteria can help to make business problems easier to solve. [1]

Example of Standard Program Header [2]:

```
/***************************************************************************
*****
                         <company/department>
***************************************************************************
*****
Program     : <Identical to file name>.SAS

Author      :

Protocol    :

Compound    :

Date        :

Purpose     :

Remarks     :

Input
- Data sets :
- Macros    :
  -Standard :
  -Study    :
- Files     :
- Format cat:
- Others    :

Output
- Data sets :
- Files     :
- Other     :

SAS version : <for which the program has been created/validated, e.g.
             windows 8.2 (winNT 4.0)
***************************************************************************
*****
(This part must be filled in for every modification that is made)
Date Modif. :
Done by     :
Description :
***************************************************************************
****/
%include sasopt;
/***************************************************************************
*****
                    FORMAT DEFINITIONS
***************************************************************************
****/
/***************************************************************************
*****
                    MACRO DEFINITIONS
***************************************************************************
****/
/***************************************************************************
*****
                     MAIN PROGRAM
***************************************************************************
****/
```

CREATE AN INTERNAL WEB PAGE OF SAS RESOURCES

This should contain (at a minimum)

- Links to sas support web site
- Links to SAS documentation
- Links to SAS-L and SAS Community
- Favorite websites (www.lexjansen.com, www.sconsig.com, www.sasprofessionals.net, among others.)

.

A SAS EXAMPLE OF WHY A MENTOR WOULD BE USEFUL

One could have an entire bookshelf of documentation just covering the subject of the Output Delivery System. Suppose as a mental exercise that one needed to train a new programmer in exclusively this subject. Would you start with (for example) the SAS Output Delivery System User's guide? That would almost certainly not be productive or useful. A gentler approach might mention that ODS was introduced in Version 6 and that it's stated purpose was to provide SAS output in many numerous ways without (it was hoped) the programmer being tied up with too many detail. My preference would be to talk about the "ODS sandwich." And build out from there.

An ODS sandwich (hold the mayo):

```
ods listing close;
ods rtf file=”blah blah blah”;
proc print data=sashelp.class;
run;
ods listing;
```

Learn how to trace (using ODS TRACE ON). Next on the list would be destinations. HTML, RTF and PDF. And, last but not least, the OUTPUT destination, which, over time will replace the familiar OUT= that all ancient SAS programmers grew up with. Learning about tagsets and templates should be on the list, as well. ODS options such as STARTPAGE=NO, BODYTITLE and STYLE= are good topics to follow[3].

A good start is to remember that ods does three main things:

- It takes raw data and combines it with several table definitions to produce the results.
- Style definitions specify background color and color scheme and the font sizes and colors of the text and numbers.
- ODS allows you to specify output destinations[4].

I find that having an outline helps to structure my learning. I realize that there are probably as many learning styles as there are SAS programmers, but my style requires having a "skeleton" to attach the bits of learning to.

MENTORING CAN BE MUTUALLY BENEFICIAL

The process of teaching other programmers (and working in close concert with) can be a learning experience for both parties. For example, a junior programmer posed the following question: Should a file be sorted before it is indexed? I had a feeling that it would be beneficial, but I wasn't completely sure. An expert in the subject[4] confirmed this hunch. Often we proceed to code in a certain fashion while only being vaguely aware as to what the underlying reasons and/or motives are.

CONCLUSION

One of the goals of mentoring should be to create a structured learning environment. Learning in such an environment should lead to an accelerated learning pattern. This may be at odds with the short term needs of a programmer's current position, but will pay many benefits later on. Looking back at my SAS training record gives new meaning to the phrase "all over the map." While I am grateful for the opportunity to have explored so many nooks and crannies of the language, it was (in hindsight) very haphazard. If I recall correctly, the "task at hand" was to convert transaction records from foreign database into a native database transaction format. My first revelation about SAS was that I could produce meaningful reports in a few hours as opposed to the weeklong (or longer!) effort required to do this in COBOL.

REFERENCES

[1] Davis, Michael. Email of Mar 31, 2011.

[2] Post on SAS-L by Quentin McMullen, Feb 15, 2004.

[3] Zender, Cynthia. Email of July 27, 2011. (a 48 inch bookshelf, not a 24 inch one, she was quick to point out).

[4]Haworth, Lauren, Cynthia Zender and Michele Burlew. *ODS: The Basics and Beyond*. Cary: SAS Press. 2009. p.14

[5]Raithel, Michael. Email of July 25, 2011.

ACKNOWLEDGMENTS

Michael Davis (a.k.a. Mad Doggy) for providing a majority of the coding tips.

Michael Raithel (a.k.a. Mikeeee!) for providing the helpful information about Westat's intranet site.

Elizabeth Axelrod for her encouragement to write this paper.

Al Verheggen for being the best IT mentor one could hope or wish for.

Peter Crawford for his many thoughtful (and thought provoking) comments and review.

RECOMMENDED READING

The two "Little SAS Books" on the SAS Language and Enterprise Guide by Delwiche and Slaughter are a good place to start, and no bookshelf worth it's salt should go without The Complete Guide(s) (SAS Macro Language and SAS Report Procedure) by Art Carpenter. The SAS Functions by Example by Ron Cody is also a constant companion.

CONTACT INFORMATION

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

Name: Stanley Fogleman
Enterprise: Harvard Clinical Research Institute
Address: 930 Commonwealth Ave West 3rd Floor
City, State ZIP: Boston MA 02215
Work Phone:
Fax:
E-mail: Sfoglemanathcridotharvarddotedu
Web:

sas Certified Advanced Programmer for SAS® 9