

Journeyman's Tools: The Writing for Reading and Reuse Program Header or Everything You Ever Wanted To Know About A Program But Were Afraid To Ask

Ronald Fehd, SAS-L's macro maven
Centers for Disease Control, and Prevention, Atlanta GA USA

ABSTRACT

Information is the difference that makes a difference. The difference between a program with documentation and one without is the cost of development. A program with documentation is more likely to be easily understood and therefore more likely to be reused thus saving the effort and expense of reinvention. This paper proposes a model program header which, like this paper, contains a title, abstract, and introduction that are designed to communicate clearly program intent and usefulness. Expected audience is intermediate to advanced users and programmers.

keywords: comments theory documentation

INTRODUCTION

What would you pay for a program with 100 SAS statements? As little as \$1000? As much as \$10,000? Having paid that bill, with that program working, would you delete it and pay another programmer to write the same functionality a second time? Would you be willing to spend another \$100 on a dozen lines of program documentation so that you wouldn't be tempted to have someone do the same work a second time?

On the last page are those dozen lines of documentation to be filled in in order to convert a program from disposable to a keeper: a reusable program.

THE MYTHICAL MAN-MONTH

Brooks [Bro95] offers some memorable thoughts which I will use below to offer for consideration an approximate dollar value for a thousand lines of code (1K lines of code: KLOC). Let us assume at the beginning that a line of code is roughly equivalent to a SAS statement.

Brooks cites several studies done in the 1960s on the number of lines of code produced per man-year while developing code for compilers and operating systems. The various studies show values from 0.5 to 10 KLOC/man-year. There are a number of factors which skew a SAS KLOC from what was measured in the midst of the previous century, most notably this quote: "Programming productivity may be increased as much as five times when a suitable high-level language is used."

BACK OF THE ENVELOPE CALCULATIONS

A typical man-year might consist of as many as 50 40-hour weeks (2000 man-hours). If we pick the middle of the range of productivity — 2–30 KLOC/year — we might guesstimate that one SAS programmer might be able to fluctuate between one and fifteen lines of code per hour. He also notes that: "Very good professional programmers are *ten times* as productive as poor ones, at same training and two-year experience level." So an order-of-magnitude productivity difference between any two programmers is not unexpected.

Just how much goes on in that hour, besides typing? Brooks divides effort into two halves: Design and Testing.

			minutes/hr	days/week	days/month
Design: 1/2	1/3	understand problem	20	1.66	7.0
		education and research			
	1/6	coding	10	0.83	3.0
Testing: 1/2	1/4	component or unit test	15	1.25	5.0
	1/4	systems or integration test	15	1.25	5.0

What are contemporary rates for SAS programmers? Let us start at dirt-cheap: \$15/hr or \$30,000/yr and go on up to experienced at \$75–\$100/hr.

You may pay from \$15 per line of inexpensive code down to less than \$6 per line of expensive code. One hundred SAS statements might cost anywhere from \$500 on up to \$2000.

Of course, these figures are ball-park figures, but they are meant to make the point that a program that is a few pages long is a significant investment, one worth an effort to preserve.

INFORMATION: THE DIFFERENCE THAT MAKES A DIFFERENCE

This example program header suggest there are two important groups of information to communicate: tell me *about* the program — program group, program type, SAS type — and tell me what to expect: purpose, description, input, process, and output.

The meta-program information is designed to communicate to the two main groups of readers of a program: users and programmers. Providing this overview creates expectations, relevance, and anchors to the reader's past experiences. In addition this facilitates a key-word search of the files in a directory.

The information about expectations is expected to convey to readers in three distinct styles what to expect.

Notes: Examples of program headers of Fehd's subroutines are available in the current SESUG Proceedings, see [macro-Array], and [J-Tools]. This material is derived from [Rhetoric], a work in progress.

CONCLUSION

Program documentation is an over-looked and under-appreciated aspect of programming effort. Effective documentation contributes to program reuse. The costs associated with documentation are miniscule in comparison to those associated with redevelopment.

REFERENCES

[Bro95] Brooks, Frederick P., Jr. (1995), *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E*, Addison-Wesley.

http://www.aw-bc.com/catalog/academic/product/0,,0201835959,00%2ben-USS_01DBC.html

<http://www.amazon.com/exec/obidos/tg/detail/-/0201835959/002-4796015-3029601?v=glance>

[macro-Array] Fehd, Ronald (1997, 2004), *%Array: construction and usage of arrays of macro variables*, Proceedings of the 22nd Annual SAS[®] Users Group International Conference, 1997. <http://www2.sas.com/proceedings/sugi22/CODERS/PAPER80.PDF>

Proceedings of the 29th Annual SAS[®] Users Group International Conference, 2004. <http://www2.sas.com/proceedings/sugi29/070-29.pdf>

Proceedings of the 12th Annual Conference of the SouthEast SAS Users Group, 2004.

[J-Tools] Fehd, Ronald (2004), *Journeyman's Tools: Two Macros to Show Calling Sequence and Parameters of Routines*,

Proceedings of the 12th Annual Conference of the SouthEast SAS Users Group, 2004.

[Rhetoric] Fehd, Ronald (2004), *A SAS[®] Companion: A Journeyman's Rhetoric of Writing for Reading and Reuse*, a Work in Progress.

Author: Ronald Fehd <mailto:Ronald.Fehd@cdc.hhs.gov>
Centers for Disease Control MS-G23 **e-mail: RJF2@cdc.gov**
4770 Buford Hwy NE
Atlanta GA 30341-3724 **bus: 770/488-8102**

End note: This paper was typeset in L^AT_EX. For further information about using L^AT_EX to write your SUG paper, consult the SAS-L archives:

<http://www.listserv.uga.edu/cgi-bin/wa?S1=sas-l>
Search for :
The subject is or contains: LaTeX
The author's address : RJF2
Since : 01 June 2003

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.

The Writing for Reading and Reuse Style Sheet: a Program Header

	<i>item</i>	<i>expected value</i>	<i>commentary</i>
	<code>/* name:</code>	<code>what</code> <code>FileName.ext</code>	
program group:	<code>when</code>	development, testing review data fix data change normal form find information summary reports	a set of related tasks
program type:	<code>which</code>	job: implements task routine: calls subroutines subroutine	to submit or execute specific purpose general purpose
SAS type:	<code>which</code>	%include without parameters %include with parameters macro procedure macro function	
purpose:	<code>why</code>		why was this written?
description:		expect this	informal: natural language prose
input:	<code>what</code>	parameters	
process:	<code>how</code>	algorithm steps	semi-formal: pseudo-code or program design language
output:	<code>where</code>	data, log, list, ODS	
notes:		side effects? uses routines or subroutines?	
usage:	<code>how</code>	submit job or typical example of use of routine or subroutine	
author:	<code>who</code>	Ronald Fehd	
change notes:	<code>who</code>	who, when, what	