



The Missing Semicolon™

Volume 12, Number 1

Winter 2010

Writing Good Program Documentation

“The process of preparing programs for a digital computer is especially attractive because it not only can be economically and scientifically rewarding, it can also be an aesthetic experience much like composing poetry or music”. Donald E. Knuth, The Art of Computer Programming, Volume 1.

Computer programs are difficult to write, test, and maintain, and in my experience, they are often not well commented or documented. This can really be a shame, since it is relatively easy to comment and to list unusual items in the program while it is fresh, instead of trying to make sense of that program sometime later when it is critical to make changes to the program.

My particular influences for this topic come from some successful Computer Scientists. Knuth sums it up well when he encourages us to take absolute artistic pride in our programs. I want my programs to look good, and that includes comments and documentation. His programs could be very complicated, but they were always documented on virtually every line.

My second influence was my first computer science professor, who required a comment on every line. Even code that did nothing needed to be commented as to why those lines were there.

At my first professional job, consultants wrote what was then called structured programming, and it was beautiful. I remember thinking that when someone looks at my programs, I want them to be as impressed as I was with the consultants’ programs.

Finally, looking at IBM source code for some of the systems modules I was involved in, I was again extremely impressed at the level of documentation and program style.

So, with at least four excellent models for documentation, I try to make my code work

well and look good. These are the programming standards we use when we develop client code.

- Consistent layout
- Initial “Flower Boxes” at the beginning of the program
- Consistent naming conventions
- Comments on every line
- Explain anomalies or strange logic
- Boxes above complicated logic or procedures
- Change logs when updates occur

I recognize that deadlines are tight, and spending the time for documentation can be difficult, but it absolutely pays off in the long run. It makes programs easy to update, change, convert, and maintain. Without this documentation, sometimes these tasks are impossible.

One other item that is important to note, is that once commenting and documentation are finished, it is critical to maintain them. If comments are wrong, they are worse than none at all.

Below is a very simple version of a program that we might write. Though it is very short, documentation principles are still shown. This program reads a file, does a bit of calculation, sorts, joins with another file, and then produces a report. Here is the undocumented version.

Continued on page 3

Look for Us as Invited Speakers at SAS Global Forum!

Choosing the Right Tool from Your SAS® and Microsoft Excel Tool Belt

By: Steven and Jennifer First



SYSTEMS SEMINAR CONSULTANTS, INC.

2997 Yarmouth Greenway Drive
Madison, WI 53711
www.sys-seminar.com
train@sys-seminar.com
1-800-997-7081

In This Issue

Writing Good Program Documentation.....Page 1
Steven First.....Page 1

President's Letter.....Page 2
Steven First.....Page 2

Automated Conversions.....Page 3

Writing Good System Documentation.....Page 4
Teresa Schudrowitz.....Page 4

Small Projects and Assessments.....Page 4

SAS Support.....Page 5

How a Skills Update Can Benefit You.....Page 6
Jennifer First.....Page 6

Quick Tips.....Page 6, 7
Rosalind Gusinow.....Page 6, 7

The Little SAS Book: Book Review.....Page 7
Jennifer First.....Page 7

SAS Training Schedule.....Page 9
Jennifer First.....Page 9

Technical Credit and Recognition.....Page 9



Letter From the President



“The real hero isn’t the guy in the \$200 suit on the corner. The real hero is the guy who gets up every morning to drive a bus, work his job, and bring home his check at the end of the week”. Paraphrased from Robert Deniro in “A Bronx Tale”.

“What does this have to do with systems?” you might ask. In our view, it is easy to emphasize the glamorous, flashy, and quick results that come from systems work. We think that documentation is one of those unsung heroes that is far too often neglected, inconsistent, or just not done because of time constraints. A small amount of effort put in up front can alleviate or avoid a great deal of cost or pain for those who have to run, maintain, and alter systems in the future.

In our experience, systems are not as well documented as they should be, and very often, mission critical assets are in a very precarious place should the wrong set of circumstances occur. A well documented system and SAS program are indeed rare.

We also maintain that documentation doesn’t take extra time but that it saves enormous time in the long run. A well-written, well-documented program or system is a wonderful piece of work that we can take enormous pride in. According to my standards, a system isn’t finished when it works the first time but rather when an average person understands it and can maintain it. We at SSC always hand our clients well-documented systems, maintaining a consistent style throughout.

With that topic in mind, we are calling this our documentation issue with an article about systems documentation and also an article on SAS program documentation. While these are very basic examples, we at SSC think that they are critical for quality systems. They are not just an add-on, if we have time. They are a necessary and worthwhile “hero” in systems development.



Steve First



Copyright © 2010 Systems Seminar Consultants, Inc. Madison, WI. All rights reserved. Printed in USA. The Missing Semicolon is a trademark of Systems Seminar Consultants, Inc. SAS, SAS/IntrNet, and SAS/ACCESS are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Consulting Services

**We Will Meet Project Objectives,
Stay On Time and Budget!**

What We Provide

- ◆ Clean, reliable, meaningful data
- ◆ Rapid time frame, with thorough attention to detail.
- ◆ Automated processes, efficiencies, innovative results

Extensive Experience

- ◆ Developing project specifications, design, coding, documentation
- ◆ Variety of industries, including: retail, finance, healthcare, marketing, manufacturing, academic, insurance, government, and telecommunications.

SAS Certified Professionals



Services

- ◆ Accessing Databases
- ◆ Analytic Systems
- ◆ Application Development
- ◆ Assessments
- ◆ Automated Conversions
- ◆ Code Review
- ◆ Dashboards
- ◆ Data Validation
- ◆ Data Cleaning
- ◆ Data Conversions
- ◆ Data Extraction & Analysis
- ◆ Data Warehousing
- ◆ Decision Support
- ◆ Dimensional Modeling
- ◆ Documentation
- ◆ Maintain SAS Systems
- ◆ Mentoring
- ◆ Platform Conversions
- ◆ Process Automation
- ◆ Program Efficiencies
- ◆ Project Management
- ◆ Reporting
- ◆ Text Processing
- ◆ Web Reporting

For assistance, recommendations, or a quote:

Email jfirst@sys-seminar.com

or Call (608) 278-9964

Writing Good Program Documentation

CONTINUED FROM PAGE 1

```
data sjfsas.partdesc;
  infile tapein;
  input partno $1-9 desc $38-67 ;
  if index(partno,2,2) in ('IN', 'MF');
proc sort data=sjfsas.partdesc; by partno;
data sjfsas.combined;
  merge sjfsas.partdesc
        sjfsas.partfile(in=onpart);
  by partno;
  if onpart;
proc contents data=sjfsas.combined;
title 'Sjfsas.Combined';run;
```

Though it could be argued that the above program works and is correct, it could be improved upon as the program below shows.

```
/* ***** */
/* SAS Warranty System      program      wtypgm10      */
/* ***** */
/* description:      */
/*      this creates a sas dataset sjfsas.partdesc from an input tape. */
/*      this dataset will be merged with other warranty data later.    */
/* ***** */
/* input:      tape from sjf systems.      */
/*      sjfsas.partfile      */
/* output:      sas dataset sjfsas.partdesc      */
/*      sas dataset sjfsas.combined      */
/* schedule:      annually      */
/* date written: 27 january 2009      */
/* author:      Steven First madison wi (608) 278-9964      */
/* revisions:      */
/* ***** */
data sjfsas.partdesc;      /* build sas ds      */
  infile tapein;      /* raw input filedef      */
  input partno $1-9      /* part number      */
        desc $38-67      /* description      */
        ;      /* end of input      */
  if index(partno,2,2) in ('IN', 'MF'); /* part internal, or mfg? */
run;      /* end of step      */

proc sort data=sjfsas.partdesc;      /* run proc sort in      */
  by partno;      /* part number order      */
run;      /* end of step      */

/* ***** */
/* join partdesc with partfile to build final combined dataset      */
/* ***** */
data sjfsas.combined;      /* build combined dataset      */
  merge sjfsas.partdesc      /* join partdesc      */
        sjfsas.partfile(in=onpart); /* and partfile      */
  by partno;      /* join key      */
  if onpart;      /* keep if on part file      */
run;      /* end of step      */

proc contents data=sjfsas.partdesc;      /* run proc contents      */
  title 'Sjfsas.Partdesc';      /* title      */
run;      /* end of step      */
/* ***** end of program wtypgm10 ***** */
```

The above documentation takes very little time to add to the program, and it should make it much easier for the next person who needs to view, modify, or update the program. It is definitely worth the extra time.



***Need a
Conversion?***

We Can Help!

Put Over 30 Years
of Experience
to Work for You

Automated Tools

Diagnostic tools
data validation
data verification

Efficiencies tools
identify expensive
jobs, steps,
instructions

SAS reads and writes
to and from virtually
any file type

Convert programs,
systems, and control
statements across
platforms:
Z/OS, UNIX, Windows

**Contact Us
for Details:**

1-800-997-7081

Writing System Documentation

Documentation is key to the successful development of a system. Distributing complete and accurate documentation provides a means to prevent and identify unofficial or unauthorized changes within the system. It provides the standard against which authorized changes are made. Therefore, as the system changes, it is vital to keep the documentation up-to-date.

While documentation is vital to a system, it can be tedious to complete or update the numerous pieces of the document. When new projects become a priority, it is often easy to leave documentation uncompleted. Unfortunately, this means documentation is often incomplete or out-of-date. Errors and omissions in the documentation can lead to errors by users and system failures, causing increased cost and disruption of workflows. Complete documentation with consistent revisions and updates will help to ensure a smooth running system. In addition, by writing documents that are easy to create and maintain, it is more likely they will be completed and kept up-to-date which will keep the systems running smoothly in the future.

System documentation is the collection of documents that describes the requirements, capabilities, limitations, design, operation, and maintenance of a system. Documentation should serve the following several requirements:

- Act as a means of communication between developers
- Serve as an information repository to be used during maintenance
- Provide information to management to help plan, budget, and schedule the development process
- Inform users on how to use and administer the system

Both narrative and graphical descriptions of the system will be used to meet these requirements.

The type of documentation and the amount of depth and detail will vary by system and the target audience. Good system documentation will include some or all of the following sections:

- Policies
- Procedures
- Specifications
- Manuals
- Reports

Policies

Policies are broadly written guidelines for conduct or action. They result from high-level decisions. When writing policies, they should be clear, indicate who has authority, and state any exceptions. It is also important to be sensitive to any “unwritten” policies.

Procedures

Procedures are subordinate to policies. They are specific statements that tell how the policies are to be carried out. Procedures should provide the logical steps by which operations are performed. They

Did You Know We Do Small Projects?

Need something done quickly?

We can help!

- ◆ Save Your Staff Time
- ◆ Quick Turn Around
- ◆ Learn Best Practices

See examples of our projects:

www.sys-seminar.com/project_resume.php

Call 1-800-997-7081 to discuss details!

should state the required action, who is to perform it, and when the action is to be performed.

Within the procedures, it is important to include the following operating procedures:

- Instructions on initiating programs
- Instructions to obtain documents required for data entry
- Instructions for entering data
- Descriptions of common error messages that may occur and how each should be handled
- Descriptions of defaults used and how to change them if necessary
- Instructions for the distribution of output

While it is important to provide written instructions, it may not always be possible to describe steps in a simple or understandable manner. In these instances, it is necessary to provide a visual representation. This may be done via screen pictures, screen layouts, or other graphics.

Continued on next page

**Confusing? Time Intensive? Expensive?
Hard to Change? Error Prone?**

Do these words describe your SAS processes?

Our experts can help with an assessment:

- ◆ Drastically improve current processes
- ◆ Custom training to improve future development

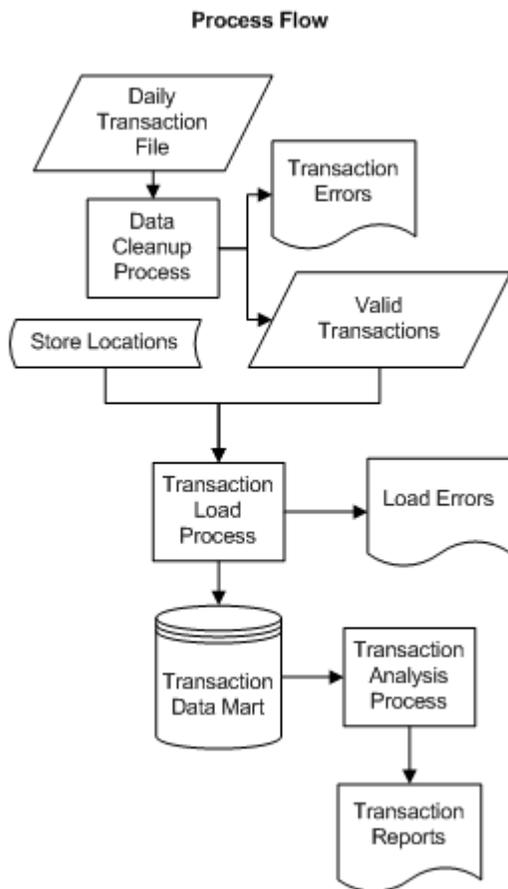
**Call for information: 1-800-997-7081 or
email consult@sys-seminar.com**

Specifications

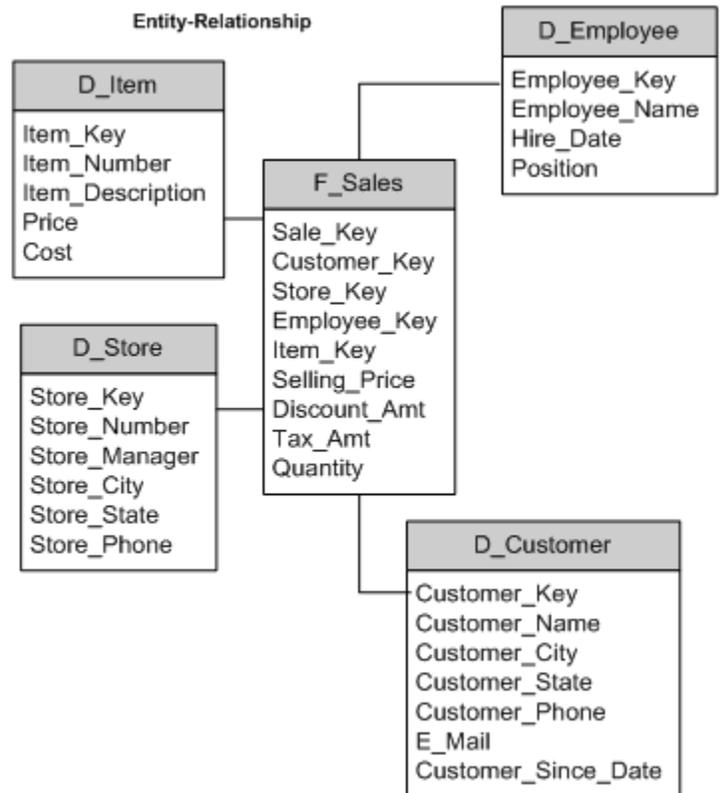
Specifications contain the technical description of the system. They are the most formal and rigid portion of the documentation. Specifications should contain the following:

- Outlines
- Flowcharts
- Data flow diagrams
- Entity-relationship diagrams
- System requirements
- Data dictionary
- Report layouts
- Test Requirements
- Input, output, and processing controls

Basic flowcharts may be included to provide a visual of the process flow:



An Entity-Relationship Diagram may be used to depict the relationships among various data structures:



Manuals

Manuals provide instructional material and are targeted towards a specific audience. There are commonly three audiences manuals are targeted for:

1. Developers
2. Operators
3. Users

The specifications provide the basis for the manuals.

Reports

Reports provide a formal communication of results and decisions. They summarize work that has been performed. Decision-oriented reports are important to provide summaries of the conclusions which have been made throughout the process.

Regardless of the size of a project, documentation is an important piece of the final system. A small project may only require a short summary of the process, an outline, and a step-by-step procedure for execution. A large enterprise system may require all of the areas outlined in this article which are comprehensive, detailed, and in-depth.

SAS Support Services

Strong Technical Support Leverages the Value of Your SAS® Software!

- ◆ Increase Efficiency and Productivity
- ◆ Custom Plan From Your Specifications

Call 1-800-997-7081 Today!



How a Skills Update Will Benefit You

Many companies and employees have been affected by our nation's economic problems and unemployment. I know many of you are worried about layoffs, or your staff is so lean that you are working twice as hard as ever and happy to be doing it. So, I suggest let's take some positive action by adding to our skills.

First, adding to your skills makes you more valuable to your current and any future employer. If you can perform at a higher level, write more efficient code, or bring new solutions to your employer, then you are the type of resource they really can't do without! So, adding to your skill-set is another form of career insurance. It makes you a stronger, more flexible, and more profitable resource. Making yourself indispensable equals job security.

Secondly, resources are tight, and everyone is working harder. A skills update will give you additional tools to handle increased responsibilities and expectations being thrust upon you as you cover a wider scope of duties. Your managers are struggling with this challenge as well. How do they meet expectations with smaller teams covering a larger range of duties? They are under a lot of pressure to maintain standards and production. This might be a good time to suggest an investment in some specialized training so that their teams can excel in this environment. This would reflect favorably on their management skills and make you more valuable!

Systems Seminar Consultants will help you move this forward. Simply ask your manager if SAS training might help your group be more effective at getting more done with less. They might respond "Yes, but our budgets have been cut". You can respond, "Well, I receive the free newsletter, *The Missing Semi-Colon*, and they are offering a 20% discount on all their offerings including on-site classes and live web classes. All you need to do is follow up with Jennifer First at (608) 278-9964 or jfirst@sys-seminar.com and mention you heard about this discount in *The Missing Semi-Colon*!"

"The biggest room in the world is the room for improvement."
- Author Unknown



Custom Training Plan

Tight Budget and Little Time?
Unsure of Department Needs?

We Will Help You Develop a
Custom Training Plan!

We Can Accomodate Diverse Needs,
Budget Concerns, and More.

Call 1-800-997-7081 Today!

QUICK TIP

A recent client required the input of CSV files they had generated. The naming convention that was agreed upon was abcdyyymmdd.csv. From the name, we could capture the create date and select the dataset that reflected the timeframe for the current processing.

Unfortunately, different individuals had different styles when naming files:

```
abcd20070331_Q1
abdef20070630_Q2
cc20070930_Q3
ddefgCJ20071007
abc2007_1014
def20071021
aaa20071028
aabCJ20071028
```

How can the date be selected from the file name when we have so many variations of the name? Use a combination of SAS functions to capture the name.

Create a SAS dataset with the dataset names from a designated library. Use the Anydigit function to find the start of the date. Create the date and continue.

Here is an example using the SAS WORK library:

```
PROC SQL;
  create table source_csv as
  select libname, memname
  from dictionary.tables
  /* sashelp.vtable */
  where libname='WORK';
QUIT;

DATA temp;
  set source_csv;
  /* remove any underscores */
  string=compress(tranwrd(memname,'_',''));
  /* grab first digit in string */
  startposition=anydigit(string);
  displayDate=input
  (substr(string,startposition,8),yymmdd8.);
RUN;

PROC PRINT data=temp;
  format displaydate date9.;
RUN;
```



Weekly Functions

This Spring, SSC will begin sending out weekly functions, with explanations and examples. Stay tuned to your email!

The Little SAS Book: Book Review

The Fourth Edition of *The Little SAS Book* is not only an excellent primer for new SAS users, but it is also a great reference and update for experienced SAS users. The content provides a comprehensive overview of the SAS System in general, and data integration, manipulation, analysis and reporting within SAS.

It begins with a basic overview of the SAS system that doesn't take much for granted, which is wonderful for a novice user who can so often feel lost when trying to break into a world of technical manuals. It also brings up details that more experienced users may have overlooked over the years just because that's the way we were taught to do things in the first place.

The Little SAS Book covers all relevant data input topics for a new SAS user and then some, from import wizards, to PC files, to raw data, to messy data. From there, it moves on to working with Data, including creating and redefining variables, SAS functions (character and numeric), subsetting, SAS dates, arrays, and more. These topics use easy to understand examples that allow any user to identify with the data and business application. The examples follow a step by step method from problem to solution that explains all of the new concepts in depth.

Of course the book also covers sorting, printing, and summarizing data. But as a big fan of web based reporting, I was really excited to see a whole chapter devoted to "Enhancing Your Output with ODS". This included not only general style options for HTML output that I found useful, but also a section on "Adding Traffic-Lighting". I am sure that every manager would love the option to develop Traffic Light reports where they could easily see when their budgets are literally "in the red".

The book included a chapter on macros that was very easy to digest for such a complex topic. It did a wonderful job of over-viewing basic macro topics, including macro variables, modular code, parameters, conditional logic, data driven programs, and macro debugging. It was a brief coverage of these topics, but the perfect amount for a brand new SAS user.

The manual overviews basic graphical and statistical procedures, exporting data to a variety of formats, and basic debugging of programs. The graphs have a few good examples of all the basic types a user would want. Examples are given for basic descriptive statistics, as well as simple regressions and analysis of variance. The "Exporting Your Data" includes other operating systems, raw data files, html, delimited files, and of course the necessary Excel file. The chapter on debugging takes a topic that can be daunting even for an experienced programmer and breaks it up into well-segmented categories. The categories make it easy for a beginner to identify common programming errors, such as the famous "missing semicolon" or missing data.

The Little SAS Book also has several valuable appendices, including

a resource section, "Coming to SAS from SPSS", and an appendix which compares SAS to other programming languages. This last section was particularly interesting to me, and would certainly help anyone who was coming from a traditional programming background to transition into SAS programming. It discusses loops, arrays, functions, procedures, data types, program structure, and compilation and execution in SAS versus other languages.

For anyone new to SAS (or those who are looking to brush up on their skills!) there is sometimes an overwhelming amount of information to absorb or sift through. *The Little SAS Book* breaks these topics down into digestible topics that offer immediate, practical application to users. Each topic covered in this book certainly deserves dozens of its own manuals, papers, and documentation, but this primer makes SAS accessible to any user. So, new users are not only ready to immediately start writing their own DATA and PROC steps, but they will also be excited to learn more about SAS in the future!



QUICK TIP

The Infile statement's MISSOVER option causes the INPUT statement to set a value to missing if the statement is unable to read an entire field because the field length that is specified in the INPUT statement is too short.

File
RAWIN

Steve 32 Monona
Tom 44
Kim 25 Madison

```
DATA ADDRESS;
  INFILE RAWIN MISSOVER;
  LENGTH CITY $10;
  INPUT NAME $ AGE CITY $;
RUN;
PROC PRINT;
  TITLE 'ADDRESS';
RUN;
```

ADDRESS			
OBS	CITY	NAME	AGE
1	Monona	Steve	32
2		Tom	44
3	Madison	Kim	25



Public and Live Web Training

Learn from the Comfort of Your Office!

Over 28 Years of SAS Training Experience

Complimentary
Follow Up
Help Desk



Our Trainers
Are Also
Expert Consultants

Upcoming Classes

Introduction to SAS®

- ◆ May 10-12
- ◆ September 13-15

SAS® Enterprise Guide

- ◆ May 13-14
- ◆ September 16-17

Advanced SAS

- ◆ March 22-24
- ◆ September 14-16

SAS® SQL Procedure

- ◆ March 16
- ◆ September 10

SAS® Report Writing

- ◆ February 22-23
- ◆ June 7-8

SAS® Macros

- ◆ April 12-13
- ◆ October 4-5

Advanced SAS® Macros

- ◆ April 14
- ◆ October 6

Exploiting SAS® ODS

- ◆ February 25-26
- ◆ June 10-11

Tips, Tricks, & SAS® Techniques

- ◆ March 25-26
- ◆ September 23-24

Introduction to PROC Report

- ◆ February 24
- ◆ June 9

SAS® Efficiencies

- ◆ April 15
- ◆ October 7

Private Training Available Upon Request

For questions or registration, contact
Our Training Coordinator at 1-800-997-7081, ext. 306
or visit www.sys-seminar.com

TECHNICAL CREDIT



Steve First
President



Jennifer First
Operations Manager



Teresa Schudrowitz
Project Manager



Rosalind Gusinow
Senior Consultant/Trainer



Timothy Broeckert
Senior Consultant



Thomas Miron
Senior Consultant