What is the SAS® System? Another View
Jay A. Jaffe

ABSTRACT

The SAS® System involves people in different ways. To explain adequately the richness of the SAS System requires empathy with people's needs. The "user" applies the SAS System to various practical problems. The "programmer" may tackle more complicated problems, assist users, or develop systems that assist users. The "manager" is concerned with utility and efficiency. This article describes an orientation approach for trainers and consultants who have occasion to introduce people to the SAS System.

INTRODUCTION

SAS consultants and trainers, and experienced SAS users who act occasionally as advisors to others, are often faced with explaining how the SAS System operates to persons who are very new to it.

Everyone who has played the novice's tutor has experienced the difficulty of summarizing what the SAS System is and what it does. The difficulty is based in the complexity of the System. And despite the SAS manuals' suggestion to the contrary, using the SAS System is not a cure for having to write applications: leaving aside those "users" who push buttons programmed by them by others (with macros, SAS/AF®, SAS/FSP®, etc.), anyone who attempts to produce good results with SAS software must have some basic understanding of it.

Not only beginners, but even those SAS users who have attained some familiarity with SAS programming can have a terrible time with this or that aspect of the SAS System. The Macro Language is often cited as a stumbling block, but there are other, sometimes seemingly small details that can throw one into a tizzy. In the DATA Step Language, there are the the esoterics of the INPUT statement, or the real way MERGE behaves, or date and time values, or the sheer mass of formats and functions. And every SAS procedure has its pitfalls, some more than others: do you know how to get PROC TABLES to work right the first time, every time? As any SAS teacher knows, the encyclopedic Basics manuals, thorough in their coverage of each detail, are quite overwhelming to beginners and even intermediate-level users looking for fundamental instruction (which is why I wrote a textbook about the basic SAS software) that takes things one step at a time.

But there are lower entry points for confusion, as far as SAS novices are concerned. SAS job flow with its stepwise structure, SAS data management and the anatomy of SAS datasets, the purposes of the SAS job log, even the very reasons for using the SAS System in the first place can be misunderstood. At one time or another, every SAS consultant, trainer, or inhouse advisor has had to explain some of the basic elements of SAS data processing.

This paper suggests an approach to teaching the most elementary aspects of the SAS System in such a way that the listener gets a firm groundwork from which to operate. Put another way, this paper presents the outline of an introductory or "pre-basics" unit, if you will, of an elementary SAS course. The approach is nothing more nor less than an attempt to tackle the question "What is the SAS System?" in a manner that speaks to the needs of the learner.

Three Types of People

It is the author's experience that in tutoring novices in the SAS System, it pays to consider who the learners are in terms of their needs. To this end, it is helpful to think of SAS novices as falling more or less into three nonexclusive categories. The "user" is well-called. S/he uses the system to meet an end. The SAS user is typically a data analyst (let's leave out the "users" who may perform data entry, etc., under guidance of an analyst), who has an application for the System. S/he may operate in a shell environment created for the purpose, such as SAS/ASSIST or a custom SAS/AF, SAS/FSP, or Macro Language driver, or else may take on simple, direct programming tasks. As s/he gains experience, s/he may act as a programmer, taking on more complex tasks when requirements emerge.

The "programmer" is a person who has a background in computer-assisted data processing, or who at least is facile and savvy with computers on some level, not afraid of the machine. Depending on his/her organizational role, or on the task at hand, the SAS programmer may act as a user ("programmer-analyst"). At other times, s/he may be more concerned with developing some special series of reports, or data processing modules, or some other system of application programs that enable users to better do their jobs. Or, s/he may be called upon to assist users directly in difficult or unusual applications.

Finally, there is the "manager", a person who directs the course of a data processing effort, broadly or in detail, and in effect decides what is important to the user or the programmer in terms of goal results. The manager may know a good deal about programming, or s/he may know very little about the technical realm.

Users, programmers, and managers bring different expectations to the SAS System. The user expects to apply it to some task. The programmer expects to write code to exact performance from it. And the manager expects to receive some tangible benefit from it, usually in the form of some printed reports. So what do each of these people need to know, right from the start? What is the SAS System to them?
Approach of This Paper

In this paper, I put forward a composite answer to this question. My answer consists of a set of facts and concepts, which I suggest as a minimum or core prescription not by any means exhaustive. There is no attempt to create a complete explanation of any topic. Rather, each topic is presented succinctly, and it is left to the reader to embellish or fill out the ideas when explaining them, as the situation warrants. Note: to distinguish them, I have italicized SAS jargon terms the first time mentioned.

NEED TO KNOW: COMMON FACTS FOR ALL SAS NOVICES
(Or, The Three-Minute Overview)

There are some central themes that anyone approaching the SAS System for the first time should grasp:

The SAS System is a collection of computer programs. Some of these are grouped under what is called the base SAS software. The base SAS software provides a compiler for a high-level programming language, the DATA Step Language, and various data-management features accessible to the compiler that help make simple some important data management tasks such as input/output, field and record subsetting, and merging.

Other SAS System programs, called SAS procedures, serve as tools a SAS user can call upon to enter, transform, and write data in well-prescribed manners. Using SAS procedures relieves the user/programmer from having to write all programs "from scratch." Instead, the user makes procedure requests that specify the data to be used and how the chosen procedures are to act upon the data. There are procedures for statistics, for manipulating and printing data, for graphics, for direct data entry, and for other tasks. Some procedures come with the base SAS software, while others are licensed in separate collections.

Because of its high-level language features and ready-to-use procedures, it is possible for novice user-programmers to get good use of the SAS System. When applications become too difficult, or when nonprogrammers are involved, an accomplished SAS programmer can build end user interfaces providing "4th generation" ease.

COMMON KNOWLEDGE FOR USERS AND PROGRAMMERS

There are some facts about which anyone wishing actually to use the SAS System needs to be aware. These concepts are not so important to managers who do not themselves plan to use the software:

SAS Datasets

SAS data is stored in special forms, the most important of which is the SAS dataset. SAS datasets are created and used by the SAS System, and not by any other software. A SAS dataset consists basically of a set of records (observations), each containing a set of data values (variables). Non-SAS data (external data) can be converted to and from SAS datasets easily, using SAS System input/output features.

The SAS Execution Environment

The base SAS software provides an environment in which SAS programs execute, called the SAS supervisor. The supervisor function intercepts SAS source code and directs the resources of the System, such as the DATA Step compiler or the various procedure modules, to respond to the demands made by the source programs.

SAS programs can be submitted for batch execution, or else data can be entered and output received at the terminal in interactive mode.

SAS Programs

All non-trivial SAS programs consist of one or more steps that are each translated and executed in turn. DATA steps invoke the DATA Step Language, and PROC steps invoke SAS procedures.

SAS jobs include as a part of their printed output a SAS log which describes the execution of the programs, files used and created, and any errors or irregularities that the System may have detected.

Besides printed output, it is possible to direct output to computer files (SAS or non-SAS) for use by other programs.

Use With Care

Because the SAS System gives the user a lot of power, it is important to check the results of programs carefully. The SAS log will tell of errors or potential errors that SAS detects, but it cannot discern many kinds of errors in logic or input data. The user should test programs carefully before relying on them.

Each SAS installation appoints a SAS software consultant who can be consulted in case of problems or questions about a SAS program or about the SAS System. This local consultant has access to SAS Institute consultants if necessary.

APPROACHING THE USER

The user wants results:

The SAS System is Task-Responsive

Because of its richness, the SAS System usually provides a choice of methods for data processing tasks. There are a great many procedures, to perform all kinds of data analyses. There are various techniques for breaking data down into categories and for printing and charting and graphing the results. As for input data, it may take almost any form that can be entered into a computer, including files from some other software packages, and the SAS System will normally be able to read it.

Appeal to the Programmer Inside the User

The motivated user can become a user-programmer: not just a buttonpusher but a
builder. As one uses the SAS System, one discovers more and more different ways to approach data analysis projects. At some point when a programmer has to apply his/her SAS knowledge to a wide variety of projects, correctly using the DATA Step Language as well as a number of procedures, the user can rightly be called a SAS programmer.

To learn the SAS System, start by studying the DATA Step Language, and practice with it and with procedures of interest. Users who have never programmed before should overcome computer fear through practice on nonessential or made-up data, and never be afraid to ask questions of more experienced SAS users, who are usually happy to help. Taking a SAS "Basics" course can also help.

**APPROACHING THE PROGRAMMER**

The programmer is interested in the distinctive technical features of a software system that promise effective --and professionally stimulating -- means to accomplish their tasks:

**The DATA Step Language**

The DATA Step Language contains expression-assignment and flow of control constructs similar to those found in other procedural languages. Its syntax has similarities to that of the PL/I language. A large set of functions for numeric, statistical, character string, and date/time computations is provided. Input and output statements provide a great deal of control over the forms external data may take.

High-level programming constructs in the DATA Step Language facilitate subsetting, interleaving, and merging datasets. All common relational-database operations can in principle be mimicked by applying these features to SAS datasets.

Explicit data declarations and certain other housekeeping associated with many procedural languages are made unnecessary by the SAS System data structures and execution environment.

**The Joys of FSP and AF**

There are SAS procedures available that make it possible to build sophisticated data entry- and verification systems, and "user-friendly" 4th-generation-appearing application front ends. Though not as easy to master as more other application builders, the underlying programs have all the power of the SAS System at their command.

**The Macro Language**

Most current implementations of the SAS System include an extremely powerful preprocessor, the SAS Macro Language, the output from which is SAS code. A SAS macro can include regular SAS code, macro variables, macro statements, and calls to macros; macros can call themselves recursively. Macro statements include several that can direct flow of control (if-then-else), simple and iterative do, goto) at preprocessing. Macro expressions can be evaluated; macro expressions consist of regular SAS code, macro variables, operators, and special macro functions.

Macro variables can include virtually any code fragment, up to a certain length. The preprocessor intercepts and expands macro variables that appear outside as well as inside macro definitions. By reduplicating the macro variable identifying token ($), delayed and indirect references are possible.

The macro preprocessor intercepts with the step-by-step flow of a SAS program by trading control with the SAS supervisor. This makes possible communication between data obtained at runtime of a previous DATA step and subsequent preprocessing. A macro can also communicate with a later DATA step. These results are achieved by special DATA step functions that read or write macro variables, or that selectively expand or call a predefined macro. Macros can also put messages and receive user-entered data, for real-time interactive applications.

**Extensions**

Under some current versions of the SAS System, and in some cases under all versions, it is possible to write specialized SAS procedures. Under some versions, it is possible to pre-compile DATA step programs and later invoke them with passed dataset names, thus protecting source code.

**APPROACHING THE MANAGER**

A manager wants to know what kinds of jobs the SAS System can help do and at what cost, and whether future as well as current needs will be met:

The SAS System provides a generalized high-level programming environment yielding flexible, responsive solutions to a broad range of data processing needs. SAS applications are especially suitable when requirements for data management and reporting change frequently, when detailed statistical analyses have to be performed, or when printed reports requirements are complex.

While the SAS System uses considerable computer resources at runtime, programmers' time to develop SAS applications can be very much less than that required by other languages or software systems. And though there exist database systems and report generators, many easier to master than SAS, few can claim the analytical power and flexibility. Furthermore, the SAS System can import and export data from and to other software systems, including databases such as IBM's DB2 on the mainframe, or dBASE and Lotus (12IP) on microcomputer.

SAS programs and datasets are portable through a wide range of machines and operating systems. The System provides features to facilitate downloading code and data between machine environments.
The question "What is the SAS System?" is often answered by an enumeration of various features of the software, with little attention paid to what people actually want from it. By contrast, I have sketched out an answer to the question that focuses on what can be done with the SAS System, not just what's in it.

It is no surprise that much of the above sounds like a sales pitch for the SAS System, for one of the elements of salesmanship is always to stress the benefits offered by a product over the objective features of the product. Helping novices learn and use the SAS System consists in the first instance of explaining them its benefits. Understanding their needs and wants focuses the discussion.

NOTES

1. From the Preface to any Version 5 or Version 6 manual: "When your computing needs are met, you are free to concentrate on results rather than on the mechanics of getting them." But in reality, the SAS System does not remove the mechanics of getting results, it just provides a different set of them.


Correspondence: Jay A. Jaffe, PhD, POB 31652, Walnut Creek, CA 94598

"SAS", "SAS/AF", and "SAS/FSM" are registered trademarks of SAS Institute, Inc.