INTRODUCTION

One of the most frustrating things about computer programming is dealing with messages or results that are unexpected. Perhaps the messages seem to be ambiguous or out of place. Perhaps the output is just not right. Presented here are some tips and guidelines to help eliminate some of the frustrations when using SAS software.

Much time and effort can be saved if a reasonable amount of energy and concentration is expended before actually beginning a project. As in "when all else fails, read the directions" so it goes in programming. The SAS System offers extreme power to all those who need calculations, statistics, graphics, and so on without requiring a great deal of background in computer science or data processing. In fact, people of any background can learn to use the SAS System effectively in a very short period of time.

BEFORE YOU START

In order to solve any problem, you must have the facts to start. This is also the case with writing a program. No matter how experienced you become, you will always need a design from which to work. Following the process below will prepare you to begin:

- Decide exactly what you want to do.
- Read the documentation.
- Look for examples.
- Know your data.
- Prepare good test data.

The amount of SAS software documentation can be overwhelming. Although it is more user friendly than most of the offerings in the field, there is much more available than you can possibly digest. Each individual document has an index to assist you. Additionally, the Master Index to SAS System Documentation Version 5 Edition and the Master Index to System Documentation for Personal Computers are very handy tools. Use them to get a starting point or a reference list by topic for your needs.

There are many places to find examples of a particular technique or the use of some of the SAS software programming statements. The two most popular places are the SAS Sample Library and the SAS Application Guide Series. The sample library contains on "INDEX" member which generates a keyword listing of the samples included. Application guides have indexes in the back of each book to guide you.

It is almost impossible to work with data that you do not understand. In order to describe the data to the SAS System in an INPUT statement, you should look at them carefully. If you cannot look at your data with an editor, you can use the SAS System as illustrated below:

```
DATA _NULL_;  
INFILE fileref;  
INPUT;  
PUT _INFILE_; 
```

The SAS System will dump the input buffer to the SAS log for you. Then you will see exactly how the raw data look and can write your INPUT statement accordingly.

Always make sure that data created for testing includes as many cases as possible. For example, if you are using BY-group processing, make sure that you have more than one BY group in the test data and multiple values of the BY variable in some groups. See what happens when the the program finds a numeric variable where a character variable is expected. Tests such as these will eliminate a lot of headaches later on.

WORK WITH THE CODE

ONE STEP AT A TIME

The modular approach to programming is a good approach because it allows you to design your logic one step at a time. It is also easier to test a few lines of code as opposed to an entire program. You can correct any problems that you encounter as the program grows. In addition, you can deal with any limitations of the operating system or the SAS System immediately. For example,
The core of the SAS System, the Program Data Vector, is limited to 32K bytes. This limit allows for approximately 4000 numeric variables. There is no circumvention within a single step. If possible, you can divide the job into more than one data step to accomplish the same results.

Another possible problem may be a lack of available memory. Each user of the operating system is allowed a fixed amount of memory. At some sites you cannot increase that limit beyond a certain point. If you are working under such a limitation, it is to your advantage to be aware of it as soon as possible.

There are some techniques that may save you coding time and machine time as well. For example, if you have certain things to do to a group of variables, you can define an array and operate on it as a unit instead of coding lots of different statements, each with the name of a particular variable. If you are doing many IF tests, perhaps you can check a range rather than a lot of individual values. If you are using a SAS data set, it may be efficient to use the POINT= and NOBS= options in the SET statement to read only selected observations rather than reading through the data set sequentially. If you are reading a large data file but only selecting a subset of the records, you may find it beneficial to read a single field or some portion of the record to test against and only read the rest of the record if necessary. This method is illustrated below:

```
DATA SUBSET;
INFILE fileref;
INPUT @1 TYPE $1. @;
IF TYPE='literal' THEN DO;
  INPUT (rest of line);
  OUTPUT;
END;
```

You may be using character variables and numeric variables together for comparisons or creating macro variables to be used in other steps. The SAS System automatically converts from one to the other if the need occurs. However, the results may be unexpected. Beware, a numeric-to-character conversion uses the BEST12 format and right-justifies the value in the receiving field. This could result in leading blanks in the front of the value that may cause problems with macro variables or in using some of the character functions such as SUBSTR.

**TIPS ON DEBUGGING**

Most programming problems will be either syntax, logic/design, system interface or limitation related. The best place to start debugging is with the SAS log. The SAS log gives information about what happened for each step in the job. It lets you know if a data set was created, and how many observations and variables it has. Always pay careful attention to all notes in the log. Error messages and warning messages are also printed here. Do not disregard the warning messages. They can be extremely helpful in figuring out exactly what is going on. Many users are interested in executing a syntax-checking run before they actually begin to use data. We do not have a particular option to allow syntax checking, but you can simulate the same thing by using the statement below:

```
OPTIONS OBS=0 NOREPLACE;
```

Note that these options have no effect on some of the procedures, and some steps in the job will run anyway. PROC DATASETS, for example, will still execute according to your code. In addition, external files are opened and closed.

One of the best debugging tools that you have available to you is the PUT statement. Used most often to monitor logic flow, the PUT statement can dump values of variables at any point or write messages to the SAS log to allow you to watch the program run.

One kind of error that occurs often is an operating system error. These errors may not appear in the SAS log. Under OS, the source of information is the JOB LOG. Under CMS, look at the FILE SYSPRINT A file to get what you need. Under minis, these notes are in the SAS log for you, and on a PC they appear on your screen. You may need documentation which is not provided by the Institute to help with these problems.

It is often helpful to use some of the SAS procedures to push you toward a solution to your problem. The CONTENTS procedure and the PRINT procedure are probably the most useful. PROC CONTENTS shows you exactly how the SAS System has defined your variables, and PROC PRINT dumps out the values you are dealing with. Use OBS=n (where n is a small number) if you just want to get an idea of what your data look like at this point.
HELP!

OK, so you have used everything you know and still cannot figure out what is going on. One of the most useful things you can do at this point is let someone else look at your program. Even someone who doesn't know the SAS language can be helpful, especially with logic problems. Programmers sometimes look at things too long until they really do not see what is there. Certainly a good approach is to use the on-site consulting that is available to you. For all mainframe and mini sites, the SAS statement

HELP SITEINFO;

lists the names of people at your site who are available to help you. For the PC system, use the SITEINFO command on the command line the same information.

Another good source of information is the Usage Notes. The Institute provides a usage note database to users that gives information on outstanding problems, usage tips, corrections to documentation, and similar kinds of things. Not all sites have these installed, but it is worth a try to check them if possible.

You may need to go back to the documentation to refresh your memory about how a feature is supposed to work. We are also in the process of publishing a new document called SAS Guide to Problem Solving and Error Messages, Version 5 Edition. This document, which will be available about mid-year, will contain information for mainframe and mini systems. It will document some of our error messages, helpful debugging tips for specific areas, and general error-resolution tips. In the meantime, believe the error message. It is most likely telling you the truth.

WHEN ALL ELSE FAILS - TECHNICAL SUPPORT

Most of you are aware that the Institute provides technical support to its installations via telephone from 9:00 to 5:00 Eastern Standard Time Monday through Friday. The service is offered to users via their SAS Software Representatives and SAS Software Consultants. When calling Technical Support, you may get a recording directing you to dial certain numbers to get the department that you need. Basically, the system is set up so that you will go in one direction if you have a new problem and another direction if you have already reported the problem to us.

When calling about a problem that has already been reported to us, always ask for second-level consulting and have the tracking/problem number ready. By having the number available, you can avoid a delay in reaching the specialist who is working on your problem and also avoid starting all over with someone else.

If you have a new problem to report, your SAS Software Consultant should ask for Technical Support first level. We need the following information:

- Site Number
- Area of Your Problem
- SAS Log
- Error Messages
- Any System Messages

We are trying a two-tier system a few hours each day. You may actually come directly to a first-level consultant rather than the Technical Support switchboard. If this is the case, the first-level consultant may be able to transfer you to a specialist in your area if he or she cannot answer your question. In either case, you need the information listed above.

We do not offer contract programming as a service, so we cannot write programs for you. Neither do we consistently have ESP, although it may seem that way sometimes. We can help you with any errors you are getting and offer application suggestions.

In spite of all we do and say for you, the best possible teacher for problem-solving techniques is experience. As you continue to program, it will continue to get easier and easier for you to "get it right the first time." The SAS System is very powerful and flexible. Remember that it is much too big for anyone to know it all. The final rule then not to give up or get more discouraged than absolutely necessary.

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA