

## Computer Validation and the SAS® System

William C. Murphy  
Howard M. Proskin & Associates, Inc., Rochester, NY

### ABSTRACT

In clinical trials, a great deal of care must be exerted to insure the integrity and safety of the database. Validation is a FDA requirement to assure the integrity of the computer hardware and software. While examining a subject, a clinician dictates scores to a recorder who enters the information into a dedicated computer. The database is transformed into the SAS system for final manipulation and analysis. Since the only mathematical changes to the data occur in the SAS system, care must be exerted to make sure the programs are validated. This requires documenting the source code and backing-up the programs in addition to developing protocol manuals and reports on the status of the database. The SAS macro facility is used to generate comments to document the source code. Another SAS routine creates a manual of the study programs. Various PROC statements are used to create status reports on the database. Back-up copies of the data sets and programs are kept onsite and offsite. Finally, the system has been examined by an outside validation expert. Using these methods, the entry, confidential storage, and analysis of clinical data comply to FDA validation guidelines.

### INTRODUCTION

If clinical trials lead to the approval of new drugs or medical procedures, the data acquired can have a profound effect on the health and welfare of the people. Even trials with negative results have the potential of denying patients the possible benefits of pharmaceuticals. With the growth and availability of computers, the data for more and more studies are acquired, maintained, and analyzed in an electronic format. To guarantee the safety of the database, the reliability of the computer system must be established. Validation is a group of procedures and requirements promulgated by the Food and Drug Administration (FDA) to assure the integrity of the computer hardware and software used in these trials. Validation is achieved by attention to five details: instruction, security, documentation, testing, and backup.

Instruction implies more than just assuring that the people operating the computers are competent. It

implies maintaining protocol manuals that explicitly state the operational details of the computer systems and descriptions of the study and database. Also in this manual should be a revision history of any changes to the protocol. Security implies limiting access to the study database to only involved individuals. This in turn involves maintaining software passwords, dedicating systems, and locking rooms. Documentation of all the software used should be readily available, not only in hard copy but in a well commented source code. The software should be thoroughly tested so that the program is doing as intended. Everything, including protocol manuals, software documentation, and the clinical trials database must be backed up regularly. This implied keeping copies of everything locked away both onsite and some distant offsite for maximum safety. Finally to assure compliance to the principles, an independent validation expert must go through a site visit. The expert should be supplied with all the manuals and documentation. He should exam the study computers environment for safety, and he should do a step by step examination of the source code. Finally he should supply the study investigators with a signed report stating his findings.

### DATA GATHERING

In our system, subjects are examined in private rooms in which an examiner dictates scores to a recorder who directly enters the information into a computer workstation. The specially designed software prevents the entry of data outside a predefined range. All such workstation are networked and the system is dedicated to a single clinical study. These test results are stored in a central file server and access to the network and the study data is double-password protected. The acquisition software was developed using state-of-the-art Windows programming tools and was carefully tested by its developers. The program is internally dated and identified and well documented. The clinics and the networked fileservers are in rooms with limited access. To make any substantive changes to the data once it is recorded a supervisor's password is also required. The data is backed up electronically to tape weekly, monthly, and quarterly, and daily in the form of printed case report forms. Back-up copies of the data sets are kept offsite and onsite within a fireproof safe.

The program has a variety of checking tools to make sure that everyone in the data base is qualified for the study. Once the study is complete, copies of the case reports forms and floppy disks of the data set are delivered to an offsite statistician for analysis. The preferred method of transfer is to hand the disk to him or his associates personally; the case report forms are mailed via some special service route. Furthermore, a variety of demographic tables are create at the clinic site to compare with the data delivered to the statisticians. This is all carefully documented to assure the consistency of data sets at the clinical site and at the statisticians' location.

## THE SAS SYSTEM

Once the statistician receives the data, he converts it into the SAS data format, where all mathematical manipulations and analyses of the data occur. Due to program speed and size considerations, the original data gathering is not done in the SAS system. The raw data is acquired in the dBase format and readily transferred using PROC DBF that is contained within SAS/ACCESS®. Since the SAS system is a major piece of commercial software that has gone through extensive alpha and beta testing and used on thousands of computer, the FDA considers the SAS system to be fully validated. However, all the programs developed in the SAS language must be documented, and backed up to insure validation of your use of the system. The various PROC statements and the macro facility of the SAS system can be used in aiding your validation of the project.

## SOFTWARE DOCUMENTATION

One of the simple thing that can be done to document your SAS programs is to develop a macro that a programmer keeps can use to put a header into the SAS program that he is writing:

```
options nonotes nosource;
data _null_;
  length project organiz invest $ 35
         comment1 comment2 $ 50;
  window Header irow=5 icolumn=10
             rows=15 Columns=65
             color=green
  #1 @3 'Project' +1 project p=no
      a=(highlight,rev_video)
  #3 @3 'Organization' +1 organiz p=no
      a=(highlight,rev_video)
  #5 @3 'Investigator' +1 invest p=no
      a=(highlight,rev_video)
  #7 @3 'Comments' +1 comment1 p=no auto=yes
      a=(highlight,rev_video)
  #8 @3 '      ' +1 comment2 p=no
```

```
      a=(highlight,rev_video);
display Header;
project=upcase(project);
organiz=upcase(organiz);
invest=upcase(invest);
comment1=upcase(comment1);
comment2=upcase(comment2);
somewhen=upcase(put(today(),worddate18.));
file 'c:\work\sas\head.tmp';
put ""
#2 'f' @;
do i=1 to 66;
  put '='@;
end;
put '
#3 ' Project: ' project @68 '|
#4 ' @68 '|
#5 ' Organization: ' organiz @68 '|
#6 ' @68 '|
#7 ' Investigator: ' invest @68 '|
#8 ' @68 '|
#9 ' Programmer: WILLIAM C. MURPHY'
    @40 'Date: ' somewhen @68 '|
#10 ' @68 '|
#11 ' Comments: ' comment1 @68 '|
#12 '          ' comment2 @68 '|
#13 ' @68 '|
#14 ' @;
do i=1 to 66;
  put '='@;
end;
put '
#15 ';;
stop;
run;
dm "include 'c:\work\sas\head.tmp';zoom on";
options notes source;
```

We have set up our system so that this code is executed when we press alt-h. A window appears that prompts us for the organization, project, and name of the investigator. It automatically stamps the program with the programmers name and the system date. It places all of this information at the top of the program in a box. Unfortunately as you work your way through the program you still must put in comments at appropriate points and you should point out any revision that you make at a future data.

## EXTERNAL DOCUMENTATION

In order to develop external documentation of your program, the SAS code can be used as input data for another program. In fact, at a previous SUGI [1], a program to read the comment lines was demonstrated. Furthermore if you follow certain program techniques, you can readily develop a program that identifies the flow of the data from one step to another. A simple

such program follows:

```
data one;
  infile instuff;
  length line $80 operatn $9 temp $15 opname library
    $8 test2 $6;
  input line &;
```

In this program, we read in the SAS program 'instuff'. Each line of the program is read into the variable 'line'. We then test the first word of the variable 'line' to see if it is a DATA STEP. If so, we scan the words after data to find the name and library of the SAS data set created. The program then checks the next line to see if it contains the words 'infile' or 'set' in order to determine the type of the input data and where it is located:

```
test=sca(line,1);
if test='DATA' then do;
  operatn='Data Step';
  temp=scan(line,2,' ');
  ntest=index(temp,' ');
  if ntest=0 then do;
    opname=trim(temp);
    library="";
  end;
  else do;
    library=scan(temp,1);
    opname=scan(temp,2);
  end;
input line&;
test2=scan(line,1);
source=scan(line,2);
if upcase(test2)='INFILE' then stype='Raw';
  else if upcase(test2)='SET' then stype='SAS';
  else stype='???';
end;
```

If the first word of the variable 'line' is not data, we then check to see if the first word is PROC. The line is then scanned to find the name of the data set that the PROC is operating on.

```
else if test='PROC' then do;
  operatn='Proc ';
  opname=scan(line,2);
  ntest=index(upcase(line),'data=');
  if ntest=0 then source='_default_';
  else source=substr(line,ntest+5);
  ntest=index(upcase(line),'data=');
  ntest=index(upcase(line),'LIBRARY=');
  if ntest=0 then library="";
  else library=compress(substr(line,ntest+8),',');
```

The format of certain PROCs requires us to look at additional lines. For example, in the case of DATASETS, we examine the 'modify' line to find out what data set the PROC is using.

```
if upcase(opname)='DATASETS' then do;
  input line &;
  ntest=scan(upcase(line),'MODIFY');
  if ntest=0 then source='???';
  else source=scan(line,2);
end;
```

Finally, if it isn't a PROC or a DATA STEP, we ignore the line:

```
else delete;
keep operatn opname source stype library;
```

A report is then printed out that lists the PROCs and DATA STEPS and their source data:

```
proc print;
  var operatn opname source stype library;
run;
```

A typical print out would look like this:

OB	OPERATN	OPNAME	SOURCE	STYPE	LIBRARY
1	Data Step	teeth	records1	Raw	library
2	Proc	datasets	teeth		library
3	Proc	sort	_default_		
4	Proc	print	_default_		

This sample shows the flow of a dental study program. The permanent data set 'library.teeth' is created from a raw data (ASCII) file called 'records1'. Three PROCs operate on this new data set.

It should be stressed that the sample program has many limitations and is in a constant state of updating. For example, as presented here, it will not deal properly with macros. Furthermore, the program assumes a certain style of program writing: the 'infile' or 'set' statement is on a line immediately after the 'data' line. However, the program can save time when you are making a flow chart for your validation documentation. Following this sample, The reader should be able to write such a program for their own documentaion. Furthermore, such a program can readily be incorporated into a macro and accessed with a key stroke as in our header macro.

## STUDY PROTOCOLS

To establish a description of our system for our protocol manual, a detailed description of our data should be included. To achieve this we have developed a series of tools. For example, to have a report on the formats of the variables we press alt-f and the following executes:

```
options nonotes nosource;
```

```

data _null_;
  /* Input SAS Format Library Path */
  length pathname $ 35 lastname $12;
  window Formats irow=8 icolumn=10
         rows=13 columns=60
         color=green
  #1 @5 'Library Path:' +1 pathname p=no
         a=(highlight,rev_video);
  display Formats;

count=length(pathname)-length(compress(pathname
,\''));
lastname=upcase(scan(pathname,count+1,\''));
call symput('name',trim(lastname));
call symput('path',compress(pathname));
stop;
run;
libname library "&path";
filename out "&path\&name..fmt";
proc printto new print=out;
title3 "Variable Formats for &name Library";
proc format fmltlib library=library;
proc printto;
run;
options notes source;

```

This program prompts for a libname and creates an output file with the format report inside. A similar program can be developed for the contents using PROC CONTENTS [2].

## DETAILS

Of course our SAS program must be backed-up and the computer on which it is used must be guarded for unwanted access. This could imply locking the room and/or computer, having a dedicated use of the computer, and extensive password protection. Finally, as done with the data gathering aspects of the study, a validation expert must examine the SAS code and the system it is used on.

## CONCLUSIONS

The SAS system can be used as a tool in its own validation. Routines can be developed that aid in the documentation of the SAS programs. Other routines can analyze the programs in order to trace the flow of data manipulation. The various procedures imbedded in macro routines can be used to described the data base for the protocol manuals. Using these tools, along with a mind toward security and an expert audit, can assure the SAS system's contribution to the clinical trial will pass validation standards.

## ACKNOWLEDGEMENTS

SAS and SAS/ACCESS are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## REFERENCES

1. Wilson, K. P. "AUTODOC: An Automated SAS Programming Documentation System", *Proceedings of the Eighteenth Annual SAS Group International Conference*, Cary, NC: SAS Institute Inc., 968-971 (1993).
2. Murphy, W. C., Proskin, H. M. and Leverett, D. H. "Tabulating the Results: An application of SAS Software to Periodic Reporting in Longitudinal Epidemiologic Study", *Proceedings of the Seventeenth Annual SAS Group International Conference*, Cary, NC: SAS Institute Inc., 1055-1058 (1992).

## CONTACT AUTHOR

William C. Murphy  
 Howard M. Proskin & Associates, Inc.  
 2468 E. Henrietta Road  
 Rochester, NY 14623

Voice (716)-235-1585  
 (716)-359-2420  
 FAX (716)-359-0465  
 eMail 71141.1651@compuserve.com