Using PROC COMPARE in a SAS/AF® Application for Tracking Corrections made to SAS® Datasets for a Clinical Trial

Emily A. Mixon, UAB, Birmingham, AL; Karen B. Fowler, UAB, Birmingham, AL

ABSTRACT

Maintaining a record of data corrections added to existing data sets is an important data management activity for clinical studies. As part of a clinical trial at The University of Alabama at Birmingham, we have created a SAS/AF® application for entering and correcting the data obtained during the trial. When corrections are made to the data the correction operator ID, the date and time the correction was made, the type of error, and the reason for the correction are recorded in the data set. Once corrections are made for the day, the user saves the data sets using an icon on the main menu of the application. The SCL behind the frame renames the data sets adding the date they were modified to the end of the original data set name and then the data sets are saved. The user then goes to the comparison portion of the application where the PROC COMPARE procedure in the SCL of the frame compares the two user selected data sets and generates output for review.

INTRODUCTION

The Department of Pediatrics, Division of Infectious Diseases at the University of Alabama at Birmingham is conducting a clinical trial where the participants in the study are followed for 42 months from enrollment. The trial started 5 years ago and enrollment is continuing at the present time. When the study first started, all changes made to the data sets by the data management personnel were logged primarily with a paper recording system. As the study progressed a more efficient and detailed account of the changes made to the data sets became necessary. In order to create an electronic trail of corrections and updates to the data sets we developed a password protected SAS/AF® Frame application in Version 8.0 of the SAS System on the Windows 2000 platform. Once the user is logged into the system, there are five choices of how to proceed from the main menu of the application (Fig. 1). The main menu frame has Icons for selections to continue in the application, a Graphic Text Control for labeling the frame and a Command Push Button for exiting the frame. The user can enter data for new study visits, make corrections or update data, save all the data sets, run a comparison of the data sets, or delete an observation. The login frame and all the choices on the main menu frame will be discussed further in this paper.

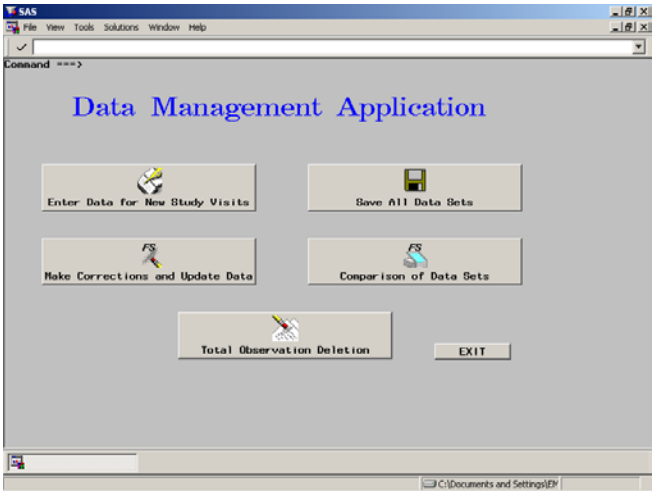

FIGURE 1

LOGIN FRAME

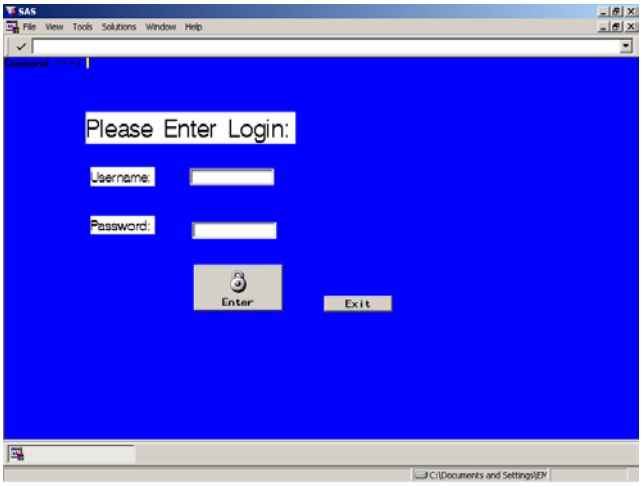

FIGURE 2

The login frame is the first screen to appear when the user selects the application icon from the desktop (Fig. 2). A username and password are entered into Text Entry Controls on the frame. The user then presses the "Enter" Icon to initiate the SCL (SCL 1). A login data set containing the username and passwords for only those data managers authorized to have access to the study data is opened in the INIT portion of the SCL. Using a WHERE statement and a FETCH function, the data set is subset based on the username and password entered. If the username and password are valid the application moves to the next frame using the CALL DISPLAY

function. If a match is not found then a message is displayed telling the user that the username and password entered are not valid. The main frame of the application is not displayed. Upon exiting the frame, the login data set is closed in the TERM portion of the SCL.

```
INIT:

DSID=OPEN('VAC.LOGINID');
RETURN;

ENTER:

RC=WHERE(DSID,'USERNAME = '|| QUOTE(USER.TEXT)
||' AND PASSWORD = '|| QUOTE(PASS.TEXT));

IF FETCH(DSID) NE -1 THEN DO;
CALL SYMPUT('USERMAC',USER.TEXT);
CALL DISPLAY('OPENINGMENU.FRAME');
END;

ELSE
_MSG_='PASSWORD INVALID FOR USERNAME GIVEN';
RETURN;

TERM:
IF DSID THEN DSID=CLOSE(DSID);
RETURN;
```

SCL 1

DATA ENTRY FOR NEW STUDY VISITS

The data entry portion of the application is coded similarly to the application described in a paper we presented at SESUG 2002 entitled "Managing Multiple SAS/FSEDIT® Data Entry with a SAS/AF® Application For a Cohort Study." Briefly, the menu frame for the data entry portion of the application contains Push Button Controls for selecting the data set needed for data entry, a Graphic Text Control for labeling the frame, and a Command Push Button for exiting the frame (Fig. 3). The SCL for the data entry menu frame uses the CALL FSEDIT routine to open the data set and FSEDIT screen selected by the user. The CALL DISPLAY routine is used to send the user to the frame for printing a listing of the data entered for the day. The SCL behind the FSEDIT screens is extensive in an effort to limit the number of data entry errors.

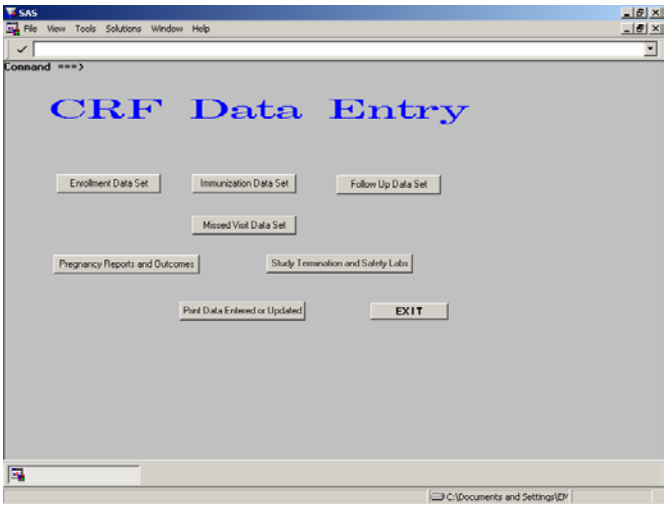

FIGURE 3

MAKE CORRECTIONS AND UPDATING THE DATA

The menu frame for the correction and update portion of the application is similar to the entry menu frame for entering new study visits. Push Button Controls are used for making a selection on the frame, a Graphic Text Control is used for labeling the frame, and a Command Push Button is used to exit the frame (Fig. 4).

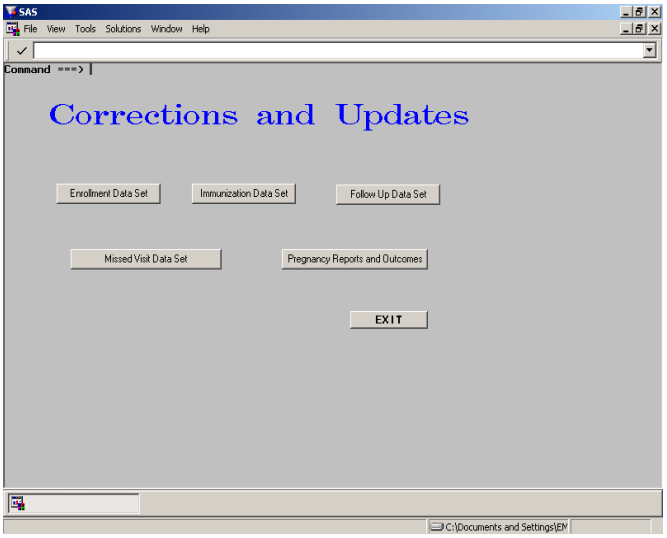

FIGURE 4

Making a selection on the frame initiates the SCL where the CALL FSEDIT routine is used for opening the appropriate data set and frame. The data set for making corrections is the same as the one for entering new data. The FSEDIT screen for the corrections has been modified to include a screen where the user fills in the error type and reason for the correction before making any changes to an observation (Fig 5).

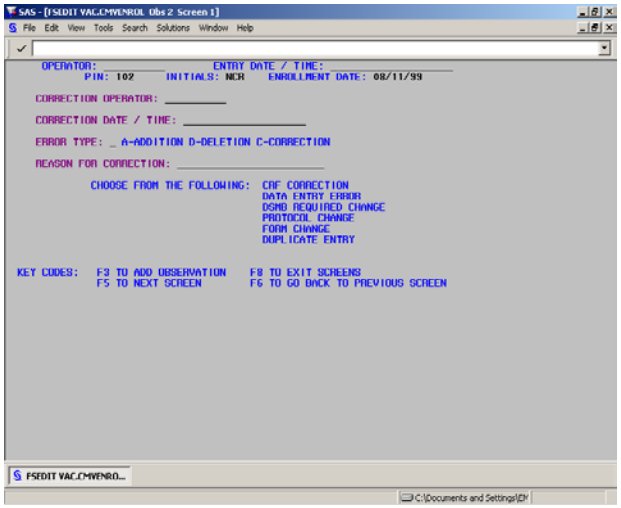

FIGURE 5

The correction operator and correction date / time fields are also required before any changes are made to an observation but this is done in the SCL of the FSEDIT screen (SCL 2). The SYMGET function fills in the protected correction operator field by using the username entered when the user logged into the application. The DATETIME function fills in the protected correction date / time field with the complete date and time. The SYMGET and DATETIME functions are initiated when the user completes the error type field. The SCL for the correction entry screens remains the same as the FSEDIT screens for entering new data.

```
IF ERROR NE ' ' THEN DO;
IF DATECORRECTED=. THEN
DATECORRECTED=DATETIME();
IF CORRECTOPERATOR=' ' THEN
CORRECTOPERATOR=SYMGET('USERMAC');
END;

IF ERROR EQ ' ' THEN DO;
IF DATECORRECTED NE . THEN DATECORRECTED=.;
IF CORRECTOPERATOR NE ' ' THEN CORRECTOPERATOR='
';
END;
```

SCL 2

TOTAL OBSERVATION DELETION

Should the user need to delete an entire observation from any of the clinical trial data sets the "Total Observation Deletion" Icon is selected from the main menu of the application. For each data set in the study there is a deleted observations data set. Therefore, should a deleted observation need to be restored it can be found in the deleted observation data set. The objects on the frame include Graphic Text Controls for labeling, Input Fields for entering the study data (i.e., PIN number, visit date or enrollment date and visit month if needed), a Radio Box for selecting the data set, an Icon for initiating the SCL on the frame, and a Command Push Button for exiting the frame (Fig 6). Once the fields are completed, the user presses the "Delete" Icon initiating the SCL behind the frame (SCL 3).

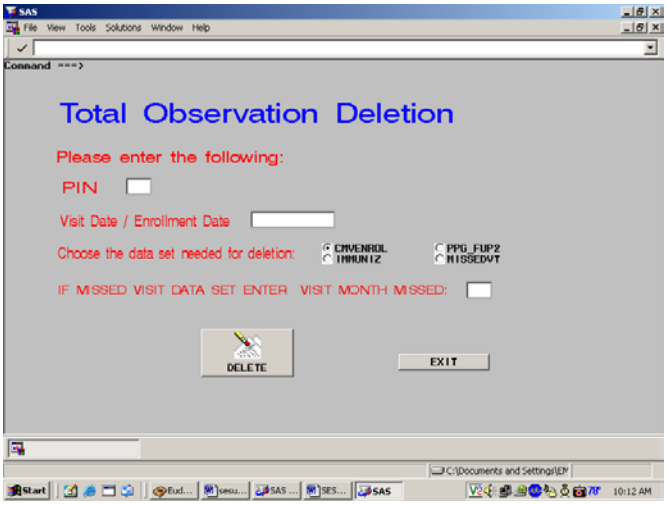

FIGURE 6

Within a SUBMIT CONTINUE routine, the data is subset by key variables (i.e., pin number, visit date, etc.) entered by the user. The correction operator and correction date / time variables are completed using the SYMGET and DATETIME functions. The observation is then appended to the appropriate deleted observation data set using the APPEND procedure. After the observation has been appended to the appropriate deleted observation data set, the data management staff manually deletes the observation from the original data set.

```
INIT:
RETURN;

DELETE:
IF DATASET IN ('CMVENROL') THEN DO;

SUBMIT CONTINUE;

LIBNAME VAC 'C:\DATA\CMVVAC';
DATA TEMP; SET VAC.CMVENROL;
WHERE ENROLDAT EQ &DATE AND PIN EQ &ID;

CORRECTOPERATOR=SYMGET('USERMAC');
DATECORRECTED=DATETIME();

PROC APPEND BASE=VAC.CMVENROL_DELETE DATA=TEMP;
RUN;

ENDSUBMIT;
END;
```

SCL 3

SAVING THE DATA SETS

Once all the data has been entered and all

corrections have been made to the data sets for the day the user proceeds to the "Save All Data Sets" Icon on the main menu of the application. The user selects the Check Box objects for the data sets that had observations added or corrected for the day and then presses the "Copy and Save" Icon (Fig. 7). Once the Icon is pressed, the SCL is initiated (SCL 4).

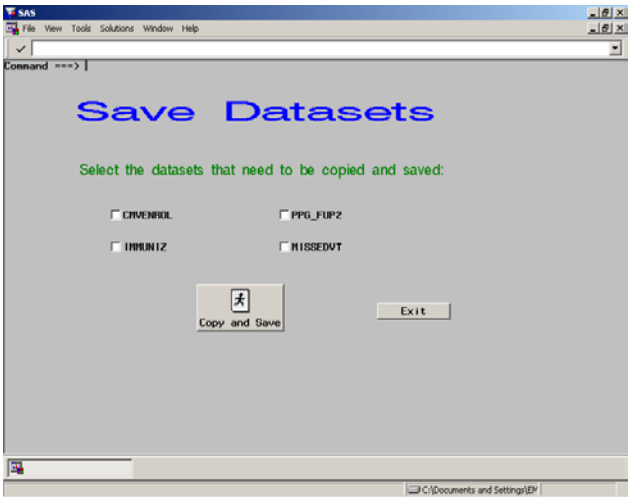

FIGURE 7

Within a SUBMIT CONTINUE routine, libnames are assigned to the folders where the data resides on the hard drive and where the data is to be saved on the server. Using PROC DATASETS the data set is selected and copied to the server. Using the dataset that was saved to the server, PROC SQL with the TRIM function renames the data set adding the current date to the end of it. The original data set remains on the hard drive of the data manager's computer with a dated copy on the server for back up and security purposes. The dated copies of the data sets are also used in providing documentation of all changes made to the data.

```
INIT:
RETURN;

SAVE:
IF BOX1 = 'E' THEN DO;

SUBMIT CONTINUE;

LIBNAME VAC 'C:\DATA\CMVVAC';
LIBNAME EM 'X:\CMVVAC\CMVENROL';

PROC DATASETS LIB=VAC MEMTYPE=DATA;
COPY OUT=EM;
SELECT CMVENROL;
RUN;

PROC SQL NOPRINT;
SELECT TRIM(MEMNAME)|| '='|| TRIM(MEMNAME)||
COMPRESS(PUT(TODAY(),WORDDATE12.-L),", ")
INTO :MEMLIST SEPARATED BY ' '
FROM DICTIONARY.TABLES
WHERE LIBNAME EQ 'EM' AND MEMNAME IN
('CMVENROL');
QUIT;
RUN;
PROC DATASETS LIBRARY=EM NOLIST;
CHANGE &MEMLIST;
QUIT;
PROC DATASETS LIB=EM;
QUIT;

ENDSUBMIT;
END;
```

SCL 4

COMPARISON OF DATA SETS

The comparison portion of the application uses the features of the COMPARE procedure to compare the differences between the same dataset on different days. By saving each clinical trial data set with the date when additions or corrections have been made we are able to generate a report of all the changes made to the original data set. When the "Comparison of Data Sets" Icon is selected from the main menu of the application the user is taken to the menu page for the comparison portion of the application (Fig. 8). A Graphic Text Control labels the frame, Desk Top Icon Controls are used for selecting the data set needed for running a comparison, and a Command Push Button allows the user to exit the frame. Once a data set is selected for comparison the user proceeds to the next frame.

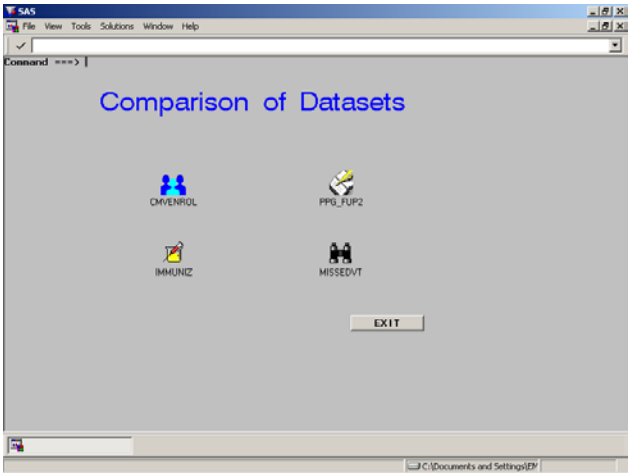

FIGURE 8

Each data set has a comparison frame. The comparison frame contains Graphic Text Control objects for labeling, List Box Controls listing all the dated data sets for the data set type selected, a Command Push Button for exiting the frame and an Icon for initiating the SCL behind the frame (Fig. 9). The List Box Controls use a SAS File List Model for displaying the appropriate data sets.

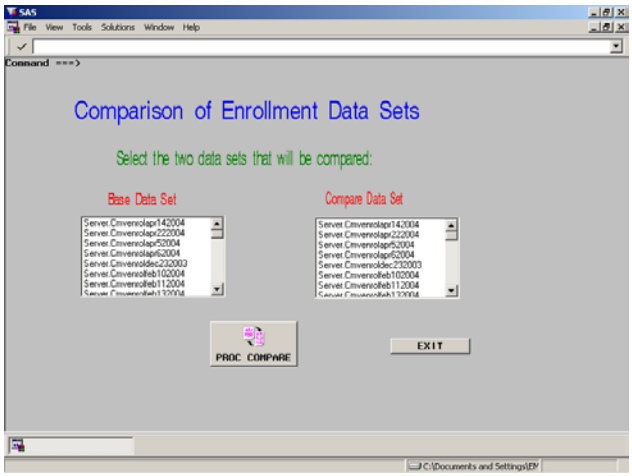

FIGURE 9

The SCL for the comparison frame begins by capturing the selected data set in each list box (SCL 5). Then, within a SUBMIT CONTINUE routine, a libname is set up for the location of the selected data. The data sets are sorted by key variables (i.e., ID number, visit date, etc.) before running PROC COMPARE. The base data set used for PROC COMPARE is the data set with the most recent date. The base data set would normally be the data set saved the day the comparison is run. The compare data set is one when data was last entered or corrected on a previous date. The COMPARE procedure compares the data set attributes, variables, variable attributes, and observations. The comparison uses key identifiers (i.e., ID, visit date, etc.) so that like observations are being compared. The OUTSTATS option is used to report the summary statistics for all the matching variables to a temporary data set with the output not printed. The temporary data set has 6 variables, however for generating our report we are interested in only three: _VAR_, _TYPE_, and _BASE_.

```
INIT:
RETURN;

COMPARE:

DATASET1=LISTBOX1.SELECTEDITEM;
DATASET2=LISTBOX2.SELECTEDITEM;

SUBMIT CONTINUE;
LIBNAME SERVER 'X:\CMVVAC\CMVENROL';

DATA A; SET &DATASET1;
PROC SORT; BY PIN ENROLDAT;

DATA B; SET &DATASET2;
PROC SORT; BY PIN ENROLDAT;

PROC COMPARE BASE=WORK.A COMPARE=WORK.B
OUTSTATS=RESULT NOPRINT;
ID PIN;
RUN;
TITLE 'PROC COMPARE OF &DATASET1 AND &DATASET2';
```

SCL 5

As the SCL for the frame continues, if the variable _TYPE_ is equal to 'NDIF' and if the variable _BASE_ is greater than 0 then the observations are kept (SCL 6). NDIF is one of the summary statistics generated during PROC COMPARE. If NDIF is the summary statistic and the corresponding _BASE_ value is greater than 0 this indicates a difference in a specific variable in our data set. The variable names from the study data sets are the values for the variable _VAR_ in the temporary data set.

```
DATA ONE; SET WORK.RESULT;
IF _TYPE_ EQ 'NDIF';
IF _BASE_ GT 0;

DATA TWO; SET ONE;
KEEP _VAR_;
DATA ADDVARS; INPUT _VAR_ $16.;
DATALINES;
PIN
;
```

SCL 6

The SCL continues with the use of the TRANSPOSE procedure (SCL 7). Another temporary data set, TRANSVAR, is created that only contains the variables from our original compare data sets that have different values. There are no observations in this data set.

```
DATA TWO; SET TWO ADDVARS;
PROC TRANSPOSE OUT=TRANSVARS;
ID _VAR_;

DATA TRANSVARS; SET TRANSVARS;
DROP _NAME_;
RUN;
```

SCL 7

A macro is then initiated for calculating the number of variables and observations in the TRANSVAR data set (SCL 8). If there were no differences found during PROC COMPARE between the two data sets a message is printed in the SAS® LOG indicating that the TRANSVAR data set contains no variables or observations. The message provides documentation that the SAS code for the comparison worked but that no discrepancies between the variables and observations were found.

```
%MACRO OBSNVARS(DS);
%GLOBAL DSET NVARS NOOBS;
%LET DSET=&DS;
%LET DSID = %SYSFUNC(OPEN(&DSET));
%IF &DSID %THEN %DO;

%LET NOOBS=%SYSFUNC(ATTRN(&DSID,NOOBS));
%LET NVARS=%SYSFUNC(ATTRN(&DSID,NVARS));
%LET RC = %SYSFUNC(CLOSE(&DSID));
```

```
%END;
%ELSE
%PUT OPEN FOR DATA SET &DSET FAILED
- %SYSFUNC(SYSMSG());
%MEND OBSNVARS;
%OBSNVARS(TRANSVARS)

%PUT &DSET HAS &NVARS VARIABLE(S) AND &NOOBS
OBSERVATION(S).;
```

SCL 8

The SCL continues with the SQL procedure where a variable list (VARLIST) is created with the variables in the TRANSVAR data set (SCL 9). The COMPARE procedure is then initiated again only this time with more options for the output and only the variables listed in the VARLIST are printed in the final report. The additional options for the COMPARE procedure are as follows: OUTNOEQUAL (to include information concerning unequal observations), OUTBASE (to provide the base data sets variables values), OUTCOMP (to provide the compare data sets variable values), and OUTDIF (to show the difference between the two observations). When reading the printed report, DIF equals "X" for a character variable indicates the observations did not match. If DIF equals a "." then the observations match for that variable. For numeric variables an "E" means the observations match, otherwise the difference between the values of the variables for the two observations will be displayed.

```
PROC SQL NOPRINT;
SELECT TRIM(NAME) INTO :VARLIST SEPARATED BY ' '
FROM DICTIONARY.COLUMNS WHERE LIBNAME='WORK' AND
MEMNAME='TRANSVARS';
QUIT;

PROC COMPARE BASE=WORK.A COMPARE=WORK.B
OUT=RESULT1 OUTNOEQUAL
OUTBASE OUTCOMP OUTDIF NOPRINT;
ID PIN;
VAR &VARLIST;

PROC PRINT DATA=RESULT1 LABEL NOOBS;
BY PIN;
ID PIN;
RUN;
ENDSUBMIT;
RC=WOUTPUT ('PRINT');
```

SCL 9

CONCLUSION

The methods discussed in this application provide our data management staff with a more extensive means of tracking corrections to study data sets. The use of PROC COMPARE and keeping dated copies of the data sets has proven to be a less complicated and an easy solution to implement in the form of a SAS/AF Frame application. The code for performing the various aspects of the renaming and saving the data sets and having only those variables with differences printed on the PROC COMPARE report may be useful in other data management tasks.

REFERENCES

Haworth, Lauren, and Karanja, Njeri. (1998), "Proving it Works: Using PROC COMPARE to Verify an Analysis Converted into SAS® Software". Proceedings of the Twenty Third Annual SAS Users Group International Conference. USA.

Mixon, Emily and Fowler, Karen. (2002), "Managing Multiple SAS/ FSEDIT® Data Entry with a SAS/AF® Application For a Cohort Study". Proceedings for the Tenth Annual Southeast SAS Users Group Conference. USA.

Ravi, Prasad. (2003), "Renaming All Variables in a SAS® Data Set Using the Information from PROC SQL's Dictionary Tables". Proceedings of the Twenty Eighth Annual SAS Users Group International Conference. USA.

SAS Institute Inc., SAS® Procedures Guide, Version 6, Third Edition, Cary, NC: SAS Institute Inc., 1990. 163 – 192, 595 – 616 pp.

SAS® Technical Support. Sample: Macro to Compute Number of Observations and Variables in a SAS Dataset. SAS Institute Inc., 2003. www.SAS.com

ACKNOWLEDGMENTS

This work was supported in part by the National Institutes of Health, National Institute of Allergy and Infectious Diseases Grant P01AI43681. The Authors would also like to thank SAS Technical Support for their helpful advice and suggestions in the development of the SCL in this application.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Emily Mixon
Suite 306, CHB
1600 7th Ave. South
Birmingham, AL 35233
Work Phone: 205-996-7792
Fax: 205-975-3221
Email: emixon@uab.edu