

Nesting SAS Macros

Marsha Weston

SRA International, Inc., San Antonio, TX

Abstract

Macros in SAS are useful for repetitive tasks. This paper illustrates “how” macros were used to scan for the presence of particular words, user-id’s, in output reports to generate a text output file outlining possible problems with HIPAA compliance.

Introduction

The Health Insurance Portability and Accountability Act of 1996 (HIPAA) had four primary objectives: (1) Assure health insurance portability by eliminating job-lock due to pre-existing medical conditions, (2) Reduce healthcare fraud and abuse, (3) Enforce standards for health information and (4) Guarantee security and privacy of health information. To be in compliance with HIPAA one must follow the Administrative Simplification (AS) provisions which detail the rules for the healthcare industry. The first rule published required implementation of Privacy and Security for healthcare organizations using any electronic means of storing patient data. Since we analyze healthcare data using stored electronic data, compliance with HIPAA was necessary.

Each week we received a report via email outlining who had accessed Privacy information from the electronic data. The report is an attached text file generated by a system using a db2 database. We needed a “quick and easy push-button” way to scan these text files and insure that we were acting in accordance with HIPAA. A SAS program was developed which allowed a “novice PC user” to enter the filenames of the email reports, the name of the users, and the name and path of an output file.

SAS Program

The SAS Program is a series of nested macros. Our SAS for Windows is version 9.1.3 SP2 under operating system, Windows 2000 Professional. The complete SAS program is in the Appendix.

User Interface, INPUT and OUTPUT

The user interface was the first part of the SAS program. Each PC user who ran the program was trained to follow the directions within the program. The user interface specifies the INPUT and OUTPUT from the program.

```
*-----;
* Name          paly2.sas              MWeston, SRA      ;
* Project        HIPAA compliance      ;
* Input          usersid, email files  ;
*      USER INPUT Follow the Steps outlined in the program ;
* Output         text file with name and date of any person in both input files;
* Date           25 September 2003    ;
* Modified                                              ;
*-----;
```

```
options mlogic mprint errors=2;
```

```

*****;
*** Step 1 Enter the IDs being scanned and the ***;
*** length of the userid. ***;
*** CAUTION! This program will only work for ***;
*** userids with a length between 6 and 9 ***;
*****;
data ids;
input @1 userids $9. @11 lenuserid 1.;
datalines;
westonma 8
smithkat 8
jonessaly 9
;
run;
*****;
*** Step 2 Enter the path and name of the output file. ***;
*** DO NOT CHANGE THE NAME OF THE LIBREF, outfile. ***;
*** the path and file name should be surrounded with quotes ***;
*** You may delete the file or it may accumulate recs. ***;
*****;
filename outfile 'C:\My Files\HIPAA\SUGI_SA\outfile.txt';

*****;
*** Step 3 Enter the File names being scanned with the path. ***;
*** Files must be saved from email as plain text ***;
*** the first period is immediately before the date. ***;
*** the path and file name should be surrounded with quotes. ***;
*****;
data in_a;
input @1 dumfname $char75.;
dbegin=scan(dumfname,2,'. ');
numrec=_n_;
datalines;
'C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt'
run;
;
run;

```

Program Logic

A. The userids that are checked for HIPAA compliance are entered in Step 1 above of the user interface. The “macro inmem” sets the attribute length of each userid variable and outputs a dataset for that length. The Log and OUTPUT windows results are intertwined below.

LOG

```

253 *-----;
254 * Name      paly2.sas              MWeston, SRA      ;
255 * Project   HIPAA compliance      ;
256 * Input     userids, email files  ;
257 *          USER INPUT Follow the Steps outlined in the program ;
258 * Output     text file with name and date of any person in both input files;
259 * Date       25 September 2003    ;
260 * Modified                                       ;

```

```

261 *-----;
262
263 options mlogic mprint errors=2;
264 *****;
265 *** Step 1 Enter the IDs being scanned and the ***;
266 *** length of the userid. ***;
267 *** CAUTION! This program will only work for ***;
268 *** userids with a length between 6 and 9 ***;
269 *****;
270 data ids;
271 input @1 userids $9. @11 lenuserid 1.;
272 datalines;

```

NOTE: The data set WORK.IDS has 3 observations and 2 variables.

NOTE: DATA statement used (Total process time):

```

real time      0.03 seconds
cpu time       0.03 seconds

```

```

276 ;
277 run;
278 *****;
279 *** Step 2 Enter the path and name of the output file. ***;
280 *** DO NOT CHANGE THE NAME OF THE LIBREF, outfile. ***;
281 *** the path and file name should be surrounded with quotes ***;
282 *** You may delete the file or it may accumulate recs. ***;
283 *****;
284 filename outfile 'C:\My Files\HIPAA\SUGI_SA\outfile.txt';
285
286 *****;
287 *** Step 3 Enter the File names being scanned with the path. ***;
288 *** Files must be saved from email as plain text ***;
289 *** the first period is immediately before the date. ***;
290 *** the path and file name should be surrounded with quotes. ***;
291 *****;
292 data in_a;
293 input @1 dumfname $char75.;
294 dbegin=scan(dumfname,2,'. ');
295 numrec=_n_;
296 datalines;

```

NOTE: The data set WORK.IN_A has 1 observations and 3 variables.

NOTE: DATA statement used (Total process time):

```

real time      0.03 seconds
cpu time       0.03 seconds

```

```

298 run;
299 title 'data in_a';
300 proc print; run;

```

NOTE: There were 1 observations read from the data set WORK.IN_A.

NOTE: PROCEDURE PRINT used (Total process time):

```

real time      0.00 seconds
cpu time       0.00 seconds

```

OUTPUT

2005	3	data in_a	09:36 Friday, May 6,
	Obs	dumfname	dbegin numrec
	1	'C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt'	092203

LOG

```

301
302 *** The SAS char functions used later only work when attrib of user name
303     is exactly the same as the user name in the text email file being scanned ***;
304 *** For each userid output a file named with the length of the userid ***;
305 data id6 id7 id8 id9;
306 set ids;
307 if lenuserid=6 then output id6;
308 else if lenuserid=7 then output id7;
309 else if lenuserid=8 then output id8;
310 else if lenuserid=9 then output id9;
311 run;

```

NOTE: There were 3 observations read from the data set WORK.IDS.

NOTE: The data set WORK.ID6 has 0 observations and 2 variables.

NOTE: The data set WORK.ID7 has 0 observations and 2 variables.

NOTE: The data set WORK.ID8 has 2 observations and 2 variables.

NOTE: The data set WORK.ID9 has 1 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time 0.09 seconds

cpu time 0.09 seconds

```

312 *** macro inmen changes the attrib of each name by length of name ***;
313 %macro inmem;
314     %do j=6 %to 9;
315         data a&j;
316             %if &j=6 %then %do;
317                 length realpers $8;
318             %end;
319             %if &j=7 %then %do;
320                 length realpers $9;
321             %end;
322             %if &j=8 %then %do;
323                 length realpers $10;
324             %end;
325             %if &j=9 %then %do;
326                 length realpers $11;
327             %end;
328             set id&j;
329             realpers=" " || trim(userids) || " ";
330             recid=_n_;
331             keep realpers recid;
332         run;
333         title "a&j";
334         proc print; run;
335     %end;
336 run;

```

```

337 %mend inmem;
338 %inmem;
MLOGIC(INMEM): Beginning execution.
MLOGIC(INMEM): %DO loop beginning; index variable J; start value is 6; stop value is 9; by
value is 1.
MPRINT(INMEM): data a6;
MLOGIC(INMEM): %IF condition &j=6 is TRUE
MPRINT(INMEM): length realpers $8;
MLOGIC(INMEM): %IF condition &j=7 is FALSE
MLOGIC(INMEM): %IF condition &j=8 is FALSE
MLOGIC(INMEM): %IF condition &j=9 is FALSE
MPRINT(INMEM): set id6;
MPRINT(INMEM): realpers="" || trim(userids) || "";
MPRINT(INMEM): recid=_n_;
MPRINT(INMEM): keep realpers recid;
MPRINT(INMEM): run;

```

NOTE: There were 0 observations read from the data set WORK.ID6.

NOTE: The data set WORK.A6 has 0 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time 0.03 seconds

cpu time 0.03 seconds

```
MPRINT(INMEM): title "a6";
```

```
MPRINT(INMEM): proc print;
```

```
MPRINT(INMEM): run;
```

NOTE: No observations in data set WORK.A6.

NOTE: PROCEDURE PRINT used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

```
MLOGIC(INMEM): %DO loop index variable J is now 7; loop will iterate again.
```

```
MPRINT(INMEM): data a7;
```

```
MLOGIC(INMEM): %IF condition &j=6 is FALSE
```

```
MLOGIC(INMEM): %IF condition &j=7 is TRUE
```

```
MPRINT(INMEM): length realpers $9;
```

```
MLOGIC(INMEM): %IF condition &j=8 is FALSE
```

```
MLOGIC(INMEM): %IF condition &j=9 is FALSE
```

```
MPRINT(INMEM): set id7;
```

```
MPRINT(INMEM): realpers="" || trim(userids) || "";
```

```
MPRINT(INMEM): recid=_n_;
```

```
MPRINT(INMEM): keep realpers recid;
```

```
MPRINT(INMEM): run;
```

NOTE: There were 0 observations read from the data set WORK.ID7.

NOTE: The data set WORK.A7 has 0 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time 0.03 seconds

cpu time 0.03 seconds

```
MPRINT(INMEM): title "a7";
```

```
MPRINT(INMEM): proc print;
```

```
MPRINT(INMEM): run;
```

NOTE: No observations in data set WORK.A7.
 NOTE: PROCEDURE PRINT used (Total process time):
 real time 0.01 seconds
 cpu time 0.01 seconds

```
MLOGIC(INMEM): %DO loop index variable J is now 8; loop will  iterate again.
MPRINT(INMEM):  data a8;
MLOGIC(INMEM): %IF condition &j=6 is FALSE
MLOGIC(INMEM): %IF condition &j=7 is FALSE
MLOGIC(INMEM): %IF condition &j=8 is TRUE
MPRINT(INMEM):  length realpers $10;
MLOGIC(INMEM): %IF condition &j=9 is FALSE
MPRINT(INMEM):  set id8;
MPRINT(INMEM):  realpers="'" || trim(userids) || "'";
MPRINT(INMEM):  recid=_n_;
MPRINT(INMEM):  keep realpers recid;
MPRINT(INMEM):  run;
```

NOTE: There were 2 observations read from the data set WORK.ID8.
 NOTE: The data set WORK.A8 has 2 observations and 2 variables.
 NOTE: DATA statement used (Total process time):
 real time 0.03 seconds
 cpu time 0.03 seconds

```
MPRINT(INMEM):  title "a8";
MPRINT(INMEM):  proc print;
MPRINT(INMEM):  run;
```

NOTE: There were 2 observations read from the data set WORK.A8.
 NOTE: PROCEDURE PRINT used (Total process time):
 real time 0.01 seconds
 cpu time 0.01 seconds

OUTPUT			a8		09:36 Friday, May 6, 2005		4
	Obs		realpers	recid			
	1		'westonma'	1			
	2		'smithkat'	2			

LOG

```
MLOGIC(INMEM): %DO loop index variable J is now 9; loop will  iterate again.
MPRINT(INMEM):  data a9;
MLOGIC(INMEM): %IF condition &j=6 is FALSE
MLOGIC(INMEM): %IF condition &j=7 is FALSE
MLOGIC(INMEM): %IF condition &j=8 is FALSE
MLOGIC(INMEM): %IF condition &j=9 is TRUE
MPRINT(INMEM):  length realpers $11;
MPRINT(INMEM):  set id9;
MPRINT(INMEM):  realpers="'" || trim(userids) || "'";
MPRINT(INMEM):  recid=_n_;
MPRINT(INMEM):  keep realpers recid;
MPRINT(INMEM):  run;
```

NOTE: There were 1 observations read from the data set WORK.ID9.
 NOTE: The data set WORK.A9 has 1 observations and 2 variables.
 NOTE: DATA statement used (Total process time):
 real time 0.03 seconds
 cpu time 0.03 seconds

MPRINT(INMEM): title "a9";
 MPRINT(INMEM): proc print;
 MPRINT(INMEM): run;

NOTE: There were 1 observations read from the data set WORK.A9.
 NOTE: PROCEDURE PRINT used (Total process time):
 real time 0.01 seconds
 cpu time 0.01 seconds

OUTPUT

a9			09:36 Friday, May 6, 2005	5
Obs	realpers	recid		
1	'joneessaly'	1		

LOG

MLOGIC(INMEM): %DO loop index variable J is now 10; loop will not iterate again.
 MPRINT(INMEM): run;
 MLOGIC(INMEM): Ending execution.

B. “Macro numobs” is used in several places to determine the number of observations in a dataset. This is a copy of the one found in the SAS MACRO documentation.

```
339 *** macro numobs determines number of obs in dataset ***;
340 %macro numobs(dsn);
341     %global num;
342     data _null_;
343         if 0 then set &dsn point=_n_ nobs=count;
344         call symput('num',left(put(count,8.)));
345     stop;
346     run;
347 %mend numobs;
```

C. “Macro cycle” is called from “Macro fileloop”. For demonstration purposes only one file is entered to be scanned for HIPAA compliance, but there is no limit to the number of files the “novice PC user” could enter. “Macro fileloop” loops through the files. It calls one file and then calls “Macro cycle” which loops through the userids entered in Step 1 above. The Log and OUTPUT windows results are intertwined below.

LOG

```
655 *** macro cycle generates a dataset for each person and then matches this ***;
656 ***     against one M2 email file                                     ***;
657 %macro cycle;
658     %do j=6 %to 9;
```

```

659      %numobs(a&j);
660      %put ***&num***;
661      %do i=1 %to &num;
662          data b&i;
663              set a&j;
664              meg=1;
665              if recid=&i then output b&i;
666          run;
667          title "b&i";
668          proc print; run;
669
670          data c&i;
671              merge one b&i;
672              by meg;
673              mat=index(string,realpers);
674              if mat > 0;
675          run;
676          title "c&i";
677          proc print; run;
678
679          %numobs(c&i);
680          %if &num > 0 %then %do;
681              data _null_;
682                  set c&i;
683                  edate=symget('ddate');
684                  file outfile mod;
685                  put @1 'Maybe Hipaa problem for' @25 realpers @37 'on' @40 edate;
686              %end;
687          %end;
688      %end;
689  %mend cycle;
690
691  %macro fileloop;
692      %numobs(in_a);
693      %put ***&num***;
694
695      %do file=1 %to &num;
696          data q&file;
697              set in_a;
698              if numrec=%eval(&file);
699          run;
700          title "q&file";
701          proc print; run;
702
703          data _null_;
704              set q&file;
705              call symput('fname',trim(dumfname));
706              call symput('ddate',trim(dbegin));
707          run;
708          %put ***&fname***;
709          %put ***&ddate***;
710
711          filename instuff "&fname";
712
713          data one;
714              infile instuff trunccover;
715              input string $char80.;

```

```

716          meg=1;
717          run;
718          title "one";
719          proc print uniform; run;
720          /* Call the macro cycle to go through the file and find userids */
721          %cycle;
722          run;
723          %end;
724 %mend fileloop;
725
726 *** call the macro fileloop to loop through it all ***;
727 %fileloop;
MLOGIC(FILELOOP): Beginning execution.
MLOGIC(NUMOBS): Beginning execution.
MLOGIC(NUMOBS): Parameter DSN has value in_a
MLOGIC(NUMOBS): %GLOBAL NUM
MPRINT(NUMOBS): data _null_;
MPRINT(NUMOBS): if 0 then set in_a point=_n_ nobs=count;
MPRINT(NUMOBS): call symput('num',left(put(count,8.)));
MPRINT(NUMOBS): stop;
MPRINT(NUMOBS): run;

NOTE: DATA statement used (Total process time):
      real time          0.01 seconds
      cpu time           0.01 seconds

MLOGIC(NUMOBS): Ending execution.
MPRINT(FILELOOP): ;
MLOGIC(FILELOOP): %PUT ***&num***
***1      ***
MLOGIC(FILELOOP): %DO loop beginning; index variable FILE; start value is 1; stop value is 1;
by value is 1.
MPRINT(FILELOOP): data q1;
MPRINT(FILELOOP): set in_a;
MPRINT(FILELOOP): if numrec=1;
MPRINT(FILELOOP): run;

NOTE: There were 1 observations read from the data set WORK.IN_A.
NOTE: The data set WORK.Q1 has 1 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time          0.01 seconds
      cpu time           0.01 seconds

MPRINT(FILELOOP): title "q1";
MPRINT(FILELOOP): proc print;
MPRINT(FILELOOP): run;
NOTE: There were 1 observations read from the data set WORK.Q1.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.01 seconds
      cpu time           0.01 seconds

```

OUTPUT:

	q1	09:36 Friday, May 6, 2005 214	
Obs	dumfname	dbegin	numrec
1	'C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt'	092203	1

LOG

```
MPRINT(FILELOOP):  data _null_;
MPRINT(FILELOOP):  set q1;
MPRINT(FILELOOP):  call symput('fname',trim(dumfname));
MPRINT(FILELOOP):  call symput('ddate',trim(dbegin));
MPRINT(FILELOOP):  run;
```

NOTE: There were 1 observations read from the data set WORK.Q1.

NOTE: DATA statement used (Total process time):

```
real time          0.01 seconds
cpu time           0.01 seconds
```

```
MLOGIC(FILELOOP):  %PUT ***&fname***
***'C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt'***
MLOGIC(FILELOOP):  %PUT ***&ddate***
***092203***
```

```
MPRINT(FILELOOP):  filename instuff "'C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt'";
MPRINT(FILELOOP):  data one;
MPRINT(FILELOOP):  infile instuff trunccover;
MPRINT(FILELOOP):  input string $char80.;
MPRINT(FILELOOP):  meg=1;
MPRINT(FILELOOP):  run;
```

NOTE: The infile INSTUFF is:

```
File Name=C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt,
RECFM=V,LRECL=256
```

NOTE: 2439 records were read from the infile INSTUFF.

The minimum record length was 0.

The maximum record length was 79.

NOTE: The data set WORK.ONE has 2439 observations and 2 variables.

NOTE: DATA statement used (Total process time):

```
real time          0.03 seconds
cpu time           0.03 seconds
```

```
MPRINT(FILELOOP):  title "one";
MPRINT(FILELOOP):  proc print uniform;
MPRINT(FILELOOP):  run;
```

NOTE: There were 2439 observations read from the data set WORK.ONE.

NOTE: PROCEDURE PRINT used (Total process time):

```
real time          0.03 seconds
cpu time           0.03 seconds
```

OUTPUT: Each line of the input file becomes one record in the SAS dataset.

one

09:36 Friday, May 6, 2005 208

Obs	string	meg
2280	act_sqlstmt SELECT	1
2281	Table__7.fy,	1
2282	Table__7.fm,	1
2283	Table__7.geo_catch_id,	1
2284		1
2285	informix_dmis63.dmis_name,	1
2286	sum(Table__7.admission_count),	1
2287		1
2288	sum(Table__7.admission_cnt_tot),	1
2289	Table__7.enrollment_status,	1
2290		1
2291	Table__7.dds,	1
2292	Table__7.sponsor_ssn,	1
2293	Table__7.sponsor_branch	1
2294	_svc,	1
2295	Table__7.ben_cat,	1
2296	sum(Table__7.total_amount_paid),	1
2297		1
2298	sum(Table__7.amount_paid_tot),	1
2299	dmis_wide5.service,	1
2300	dmis_wide	1
2301	5.parent_dmis_id,	1
2302	dmis_wide5.parent_dmis_name,	1
2303	informix_dmis	1
2304	63.service	1
2305	FROM	1
2306	'informix'.hcsr_i Table__7,	1
2307	OUTER 'informi	1
2308	x'.dmis_wide dmis_wide5,	1
2309	OUTER 'informix'.dmis informix_dmis6	1
2310	3	1
2311	WHERE	1
2312	(Table__7.enr_dmis_id=dmis_wide5.dmis_id)	1
2313	AND (	1
2314	Table__7.fy=dmis_wide5.fy)	1
2315	AND (informix_dmis63.fy=Table__	1
2316	7.fy)	1
2317	AND (informix_dmis63.dmis_id=Table__7.geo_catch_id)	1
2318		1
2319	AND ('xyzy:' 'westonma' = 'xyzy:' 'westonma')	1
2320	AND (	1
2321		1
2322	Table__7.fy IN (2002)	1
2323	AND informix_dmis63.service = 'A	1
2324	,	1
2325	AND Table__7.fm > 6	1
2326	)	1
2327	GROUP BY	1
2328	1,	1
2329	2,	1

2330	3,	1
2331		1
2332	4,	1
2333	7,	1
2334	8,	1
2335	9,	1
2336	10,	1
2337	11,	1
2338	14,	1
2339	15,	1
2340	16,	1
2341		1
2342	17	1
2343		1
2344		1
2345	act_endtime 2003-09-22 16:06:04.81613	1

LOG

MLOGIC(CYCLE): Beginning execution.
MLOGIC(CYCLE): %D0 loop beginning; index variable J; start value is 6; stop value is 9; by value is 1.

MLOGIC(NUMOBS): Beginning execution.
MLOGIC(NUMOBS): Parameter DSN has value a6
MLOGIC(NUMOBS): %GLOBAL NUM
MPRINT(NUMOBS): data _null_;
MPRINT(NUMOBS): if 0 then set a6 point=_n_ nob=count;
MPRINT(NUMOBS): call symput('num',left(put(count,8.)));
MPRINT(NUMOBS): stop;
MPRINT(NUMOBS): run;

NOTE: DATA statement used (Total process time):
real time 0.00 seconds
cpu time 0.00 seconds

MLOGIC(NUMOBS): Ending execution.
MPRINT(CYCLE): ;
MLOGIC(CYCLE): %PUT ***&num***
0

MLOGIC(CYCLE): %D0 loop beginning; index variable I; start value is 1; stop value is 0; by value is 1. Loop will not be executed.

MLOGIC(CYCLE): %D0 loop index variable J is now 7; loop will iterate again.

MLOGIC(NUMOBS): Beginning execution.
MLOGIC(NUMOBS): Parameter DSN has value a7
MLOGIC(NUMOBS): %GLOBAL NUM
MPRINT(NUMOBS): data _null_;
MPRINT(NUMOBS): if 0 then set a7 point=_n_ nob=count;
MPRINT(NUMOBS): call symput('num',left(put(count,8.)));
MPRINT(NUMOBS): stop;
MPRINT(NUMOBS): run;

NOTE: DATA statement used (Total process time):
real time 0.00 seconds
cpu time 0.00 seconds

```

MLOGIC(NUMOBS): Ending execution.
MPRINT(CYCLE): ;
MLOGIC(CYCLE): %PUT ***&num***
***0      ***
MLOGIC(CYCLE): %DO loop beginning; index variable I; start value is 1; stop value is 0; by
              value is 1. Loop will not be executed.
MLOGIC(CYCLE): %DO loop index variable J is now 8; loop will iterate again.
MLOGIC(NUMOBS): Beginning execution.
MLOGIC(NUMOBS): Parameter DSN has value a8
MLOGIC(NUMOBS): %GLOBAL NUM
MPRINT(NUMOBS): data _null_;
MPRINT(NUMOBS): if 0 then set a8 point=_n_ nobs=count;
MPRINT(NUMOBS): call symput('num',left(put(count,8.)));
MPRINT(NUMOBS): stop;
MPRINT(NUMOBS): run;

```

```

NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

```

```

MLOGIC(NUMOBS): Ending execution.
MPRINT(CYCLE): ;
MLOGIC(CYCLE): %PUT ***&num***
***2      ***
MLOGIC(CYCLE): %DO loop beginning; index variable I; start value is 1; stop value is 2; by
              value is 1.
MPRINT(CYCLE): data b1;
MPRINT(CYCLE): set a8;
MPRINT(CYCLE): meg=1;
MPRINT(CYCLE): if recid=1 then output b1;
MPRINT(CYCLE): run;

```

```

NOTE: There were 2 observations read from the data set WORK.A8.
NOTE: The data set WORK.B1 has 1 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time          0.03 seconds
      cpu time           0.03 seconds

```

```

MPRINT(CYCLE): title "b1";
MPRINT(CYCLE): proc print;
MPRINT(CYCLE): run;
NOTE: There were 1 observations read from the data set WORK.B1.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.01 seconds
      cpu time           0.01 seconds

```

OUTPUT

b1				09:36 Friday, May 6, 2005 263
Obs	realpers	recid	meg	
1	'westonma'	1	1	

LOG

```
MPRINT(CYCLE):  data c1;
MPRINT(CYCLE):  merge one b1;
MPRINT(CYCLE):  by meg;
MPRINT(CYCLE):  mat=index(string,realpers);
MPRINT(CYCLE):  if mat > 0;
MPRINT(CYCLE):  run;
```

NOTE: There were 2439 observations read from the data set WORK.ONE.

NOTE: There were 1 observations read from the data set WORK.B1.

NOTE: The data set WORK.C1 has 6 observations and 5 variables.

NOTE: DATA statement used (Total process time):

real time	0.03 seconds
cpu time	0.03 seconds

```
MPRINT(CYCLE):  title "c1";
MPRINT(CYCLE):  proc print;
MPRINT(CYCLE):  run;
```

NOTE: There were 6 observations read from the data set WORK.C1.

NOTE: PROCEDURE PRINT used (Total process time):

real time	0.00 seconds
cpu time	0.00 seconds

OUTPUT

		c1	09:36 Friday, May 6, 2005 264			
Obs		string	meg	realpers	recid	mat
1	AND	('xyzyz:' 'westonma' = 'xyzyz:' 'westonma')	1	'westonma'	1	21
2		urnerm' = 'xyzyz:' 'westonma')	1	'westonma'	1	36
3	AND	('xyzyz:' 'westonma' = 'xyzyz:' 'westonma')	1	'westonma'	1	21
4		urnerm' = 'xyzyz:' 'westonma')	1	'westonma'	1	36
5	AND	('xyzyz:' 'westonma' = 'xyzyz:' 'westonma')	1	'westonma'	1	21
6	AND	('xyzyz:' 'westonma' = 'xyzyz:' 'westonma')	1	'westonma'	1	21

LOG

```
MLOGIC(NUMOBS): Beginning execution.
MLOGIC(NUMOBS): Parameter DSN has value c1
MLOGIC(NUMOBS): %GLOBAL NUM
MPRINT(NUMOBS):  data _null_;
MPRINT(NUMOBS):  if 0 then set c1 point=_n_ nobs=count;
MPRINT(NUMOBS):  call symput('num',left(put(count,8.)));
MPRINT(NUMOBS):  stop;
MPRINT(NUMOBS):  run;
```

NOTE: DATA statement used (Total process time):

real time	0.00 seconds
cpu time	0.00 seconds

MLOGIC(NUMOBS): Ending execution.

```

MPRINT(CYCLE): ;
MLOGIC(CYCLE): %IF condition &num > 0 is TRUE
MPRINT(CYCLE): data _null_;
MPRINT(CYCLE): set c1;
MPRINT(CYCLE): edate=symget('ddate');
MPRINT(CYCLE): file outfile mod;
MPRINT(CYCLE): put @1 'Maybe Hipaa problem for' @25 realpers @37 'on' @40 edate;
MLOGIC(CYCLE): %DO loop index variable I is now 2; loop will iterate again.

```

NOTE: The file OUTFILE is:
 File Name=C:\My Files\HIPAA\SUGI_SA\outfile.txt,
 RECFM=V,LRECL=256

NOTE: 6 records were written to the file OUTFILE.
 The minimum record length was 45.
 The maximum record length was 45.
 NOTE: There were 6 observations read from the data set WORK.C1.
 NOTE: DATA statement used (Total process time):
 real time 0.01 seconds
 cpu time 0.00 seconds

```

MPRINT(CYCLE): data b2;
MPRINT(CYCLE): set a8;
MPRINT(CYCLE): meg=1;
MPRINT(CYCLE): if recid=2 then output b2;
MPRINT(CYCLE): run;

```

NOTE: There were 2 observations read from the data set WORK.A8.
 NOTE: The data set WORK.B2 has 1 observations and 3 variables.
 NOTE: DATA statement used (Total process time):
 real time 0.01 seconds
 cpu time 0.01 seconds

```

MPRINT(CYCLE): title "b2";
MPRINT(CYCLE): proc print;
MPRINT(CYCLE): run;
NOTE: There were 1 observations read from the data set WORK.B2.
NOTE: PROCEDURE PRINT used (Total process time):
      real time            0.00 seconds
      cpu time             0.00 seconds

```

OUTPUT				
b2				09:36 Friday, May 6, 2005 265
Obs	realpers	recid	meg	
1	'smithkat'	2	1	

LOG

```

MPRINT(CYCLE): data c2;
MPRINT(CYCLE): merge one b2;
MPRINT(CYCLE): by meg;
MPRINT(CYCLE): mat=index(string,realpers);
MPRINT(CYCLE): if mat > 0;

```

```
MPRINT(CYCLE):    run;
```

NOTE: There were 2439 observations read from the data set WORK.ONE.

NOTE: There were 1 observations read from the data set WORK.B2.

NOTE: The data set WORK.C2 has 0 observations and 5 variables.

NOTE: DATA statement used (Total process time):

```
    real time          0.01 seconds
    cpu time           0.01 seconds
```

```
MPRINT(CYCLE):    title "c2";
```

```
MPRINT(CYCLE):    proc print;
```

```
MPRINT(CYCLE):    run;
```

NOTE: No observations in data set WORK.C2.

NOTE: PROCEDURE PRINT used (Total process time):

```
    real time          0.01 seconds
    cpu time           0.01 seconds
```

```
MLOGIC(NUMOBS):   Beginning execution.
```

```
MLOGIC(NUMOBS):   Parameter DSN has value c2
```

```
MLOGIC(NUMOBS):   %GLOBAL  NUM
```

```
MPRINT(NUMOBS):   data _null_;
```

```
MPRINT(NUMOBS):   if 0 then set c2 point=_n_ nobs=count;
```

```
MPRINT(NUMOBS):   call symput('num',left(put(count,8.)));
```

```
MPRINT(NUMOBS):   stop;
```

```
MPRINT(NUMOBS):   run;
```

NOTE: DATA statement used (Total process time):

```
    real time          0.00 seconds
    cpu time           0.00 seconds
```

```
MLOGIC(NUMOBS):   Ending execution.
```

```
MPRINT(CYCLE):    ;
```

```
MLOGIC(CYCLE):    %IF condition &num > 0 is FALSE
```

```
MLOGIC(CYCLE):    %DO loop index variable I is now 3; loop will not iterate again.
```

```
MLOGIC(CYCLE):    %DO loop index variable J is now 9; loop will  iterate again.
```

```
MLOGIC(NUMOBS):   Beginning execution.
```

```
MLOGIC(NUMOBS):   Parameter DSN has value a9
```

```
MLOGIC(NUMOBS):   %GLOBAL  NUM
```

```
MPRINT(NUMOBS):   data _null_;
```

```
MPRINT(NUMOBS):   if 0 then set a9 point=_n_ nobs=count;
```

```
MPRINT(NUMOBS):   call symput('num',left(put(count,8.)));
```

```
MPRINT(NUMOBS):   stop;
```

```
MPRINT(NUMOBS):   run;
```

NOTE: DATA statement used (Total process time):

```
    real time          0.00 seconds
    cpu time           0.00 seconds
```

```
MLOGIC(NUMOBS):   Ending execution.
```

```
MPRINT(CYCLE):    ;
```

```
MLOGIC(CYCLE):    %PUT ***&num***
```

```
***1      ***
```

```
MLOGIC(CYCLE): %DO loop beginning; index variable I; start value is 1; stop value is 1; by
               value is 1.
```

```
MPRINT(CYCLE): data b1;
MPRINT(CYCLE): set a9;
MPRINT(CYCLE): meg=1;
MPRINT(CYCLE): if recid=1 then output b1;
MPRINT(CYCLE): run;
```

NOTE: There were 1 observations read from the data set WORK.A9.

NOTE: The data set WORK.B1 has 1 observations and 3 variables.

NOTE: DATA statement used (Total process time):

```
real time      0.03 seconds
cpu time       0.03 seconds
```

```
MPRINT(CYCLE): title "b1";
MPRINT(CYCLE): proc print;
MPRINT(CYCLE): run;
```

NOTE: There were 1 observations read from the data set WORK.B1.

NOTE: PROCEDURE PRINT used (Total process time):

```
real time      0.00 seconds
cpu time       0.00 seconds
```

OUTPUT

b1				09:36 Friday, May 6, 2005 266
Obs	realpers	recid	meg	
1	'jonessaly'	1	1	

LOG

```
MPRINT(CYCLE): data c1;
MPRINT(CYCLE): merge one b1;
MPRINT(CYCLE): by meg;
MPRINT(CYCLE): mat=index(string,realpers);
MPRINT(CYCLE): if mat > 0;
MPRINT(CYCLE): run;
```

NOTE: There were 2439 observations read from the data set WORK.ONE.

NOTE: There were 1 observations read from the data set WORK.B1.

NOTE: The data set WORK.C1 has 2 observations and 5 variables.

NOTE: DATA statement used (Total process time):

```
real time      0.01 seconds
cpu time       0.01 seconds
```

```
MPRINT(CYCLE): title "c1";
MPRINT(CYCLE): proc print;
MPRINT(CYCLE): run;
```

NOTE: There were 2 observations read from the data set WORK.C1.

NOTE: PROCEDURE PRINT used (Total process time):

```
real time      0.00 seconds
cpu time       0.00 seconds
```

OUTPUT

	c1	09:36 Friday, May 6, 2005 267			
Obs	string	meg	realpers	recid	mat
1	('xyzzzy:' 'jonessaly' = 'xyzzzy:' 'jonessaly')	1	'jonessaly'	1	28
2	ym' = 'xyzzzy:' 'jonessaly')	1	'jonessaly'	1	3

LOG

```
MLOGIC(NUMOBS): Beginning execution.
MLOGIC(NUMOBS): Parameter DSN has value c1
MLOGIC(NUMOBS): %GLOBAL NUM
MPRINT(NUMOBS): data _null_;
MPRINT(NUMOBS): if 0 then set c1 point=_n_ nobs=count;
MPRINT(NUMOBS): call symput('num',left(put(count,8.)));
MPRINT(NUMOBS): stop;
MPRINT(NUMOBS): run;
```

```
NOTE: DATA statement used (Total process time):
      real time          0.01 seconds
      cpu time           0.01 seconds
```

```
MLOGIC(NUMOBS): Ending execution.
MPRINT(CYCLE): ;
MLOGIC(CYCLE): %IF condition &num > 0 is TRUE
MPRINT(CYCLE): data _null_;
MPRINT(CYCLE): set c1;
MPRINT(CYCLE): edate=symget('ddate');
MPRINT(CYCLE): file outfile mod;
MPRINT(CYCLE): put @1 'Maybe Hipaa problem for' @25 realpers @37 'on' @40 edate;
MLOGIC(CYCLE): %DO loop index variable I is now 2; loop will not iterate again.
MLOGIC(CYCLE): %DO loop index variable J is now 10; loop will not iterate again.
MLOGIC(CYCLE): Ending execution.
MPRINT(FILELOOP): ;
MPRINT(FILELOOP): run;
```

```
NOTE: The file OUTFILE is:
      File Name=C:\My Files\HIPAA\SUGI_SA\outfile.txt,
      RECFM=V,LRECL=256
```

```
NOTE: 2 records were written to the file OUTFILE.
      The minimum record length was 45.
      The maximum record length was 45.
NOTE: There were 2 observations read from the data set WORK.C1.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds
```

```
MLOGIC(FILELOOP): %DO loop index variable FILE is now 2; loop will not iterate again.
MLOGIC(FILELOOP): Ending execution.
728
729 /*
730 proc datasets lib=work memtype=data kill; run; quit;
```

The output file, “File Name=C:\My Files\HIPAA\SUGI_SA\outfile.txt”

Maybe Hipaa problem for 'westonma' on 092203
 Maybe Hipaa problem for 'westonma' on 092203
 Maybe Hipaa problem for 'westonma' on 092203
 Maybe Hipaa problem for 'westonma' on 092203
 Maybe Hipaa problem for 'westonma' on 092203
 Maybe Hipaa problem for 'westonma' on 092203
 Maybe Hipaa problem for 'jonessaly' on 092203
 Maybe Hipaa problem for 'jonessaly' on 092203

Summary/Conclusion

A SAS program, an example of using nesting macros, allowed our project to insure compliance with HIPAA.

References

SAS[®] Macro Programming: Advanced Topics Course Notes
 SAS[®] Guide to Macro Processing, Version 6, Second Edition

Trademark Information

SAS and all other SAS Institute, Inc. product or service name are registered trademarks or trademarks of SAS Institute, Inc in the USA and other countries.

About the Author

The author welcomes your comments and suggestions.

Marsha Weston
 SRA International, Inc.
 1777 NE Loop 410, Suite 510
 San Antonio, TX 78217
 (210) 832-5243
marsha_weston@sra.com (underscore between first and last name)
www.sra.com

Appendix

```

*-----;
* Name      paly2.sas              MWeston, SRA      ;
* Project    HIPAA compliance      ;
* Input      userids, email files  ;
*      USER INPUT  Follow the Steps outlined in the program ;
* Output     text file with name and date of any person in both input files;
* Date       25 September 2003    ;
* Modified   ;
*-----;

options mlogic mprint errors=2;
*****;
*** Step 1 Enter the IDs being scanned and the ***;
***      length of the userid. ***;
*** CAUTION! This program will only work for ***;
***      userids with a length between 6 and 9 ***;
*****;
data ids;
input @1 userids $9. @11 lenuserid 1.;
datalines;
westonma 8
smithkat 8
jonessaly 9
;
run;
*****;
*** Step 2 Enter the path and name of the output file. ***;
*** DO NOT CHANGE THE NAME OF THE LIBREF, outfile. ***;
*** the path and file name should be surrounded with quotes ***;
*** You may delete the file or it may accumulate recs. ***;
*****;
filename outfile 'C:\My Files\HIPAA\SUGI_SA\outfile.txt';

*****;
*** Step 3 Enter the File names being scanned with the path. ***;
*** Files must be saved from email as plain text ***;
*** the first period is immediately before the date. ***;
*** the path and file name should be surrounded with quotes. ***;
*****;
data in_a;
input @1 dumfname $char75.;
dbegin=scan(dumfname,2,'. ');
numrec=_n_;
datalines;
'C:\My Files\HIPAA\SUGI_SA\092203.sql.pa.txt'
run;
title 'data in_a';
proc print; run;

*** The SAS char functions used later only work when attrib of user name
      is exactly the same as the user name in the text email file being scanned ***;
*** For each userid output a file named with the length of the userid ***;
data id6 id7 id8 id9;

```

```

set ids;
if lenuserid=6 then output id6;
else if lenuserid=7 then output id7;
else if lenuserid=8 then output id8;
else if lenuserid=9 then output id9;
run;
*** macro inmem changes the attrib of each name by length of name ***;
%macro inmem;
  %do j=6 %to 9;
    data a&j;
      %if &j=6 %then %do;
        length realpers $8;
      %end;
      %if &j=7 %then %do;
        length realpers $9;
      %end;
      %if &j=8 %then %do;
        length realpers $10;
      %end;
      %if &j=9 %then %do;
        length realpers $11;
      %end;
      set id&j;
      realpers="" || trim(userids) || "";
      recid=_n_;
      keep realpers recid;
    run;
    title "a&j";
    proc print; run;
  %end;
run;
%mend inmem;
%inmem;

*** macro numobs determines number of obs in dataset ***;
%macro numobs(dsn);
  %global num;
  data _null_;
    if 0 then set &dsn point=_n_ nobs=count;
    call symput('num',left(put(count,8.)));
    stop;
  run;
%mend numobs;

*** macro cycle generates a dataset for each person and then matches this ***;
***   against one M2 email file                                     ***;
%macro cycle;
  %do j=6 %to 9;
    %numobs(a&j);
    %put ***&num***;
    %do i=1 %to &num;
      data b&i;
        set a&j;
        meg=1;
        if recid=&i then output b&i;
      run;
      title "b&i";
    %end;
  %end;

```

```

proc print; run;

data c&i;
  merge one b&i;
  by meg;
  mat=index(string,realpers);
  if mat > 0;
run;
title "c&i";
proc print; run;

%numobs(c&i);
%if &num > 0 %then %do;
  data _null_;
    set c&i;
    edate=symget('ddate');
    file outfile mod;
    put @1 'Maybe Hipaa problem for' @25 realpers @37 'on' @40 edate;
  %end;
%end;
%end;
%mend cycle;

%macro fileloop;
  %numobs(in_a);
  %put ***&num***;

  %do file=1 %to &num;
    data q&file;
      set in_a;
      if numrec=%eval(&file);
    run;
    title "q&file";
    proc print; run;

    data _null_;
      set q&file;
      call symput('fname',trim(dumfname));
      call symput('ddate',trim(dbegin));
    run;
    %put ***&fname***;
    %put ***&ddate***;

    filename instuff "&fname";

    data one;
      infile instuff trunccover;
      input string $char80.;
      meg=1;
    run;
    title "one";
    proc print uniform; run;
    %* Call the macro cycle to go through the file and find userids *;
    %cycle;
    run;
  %end;
%mend fileloop;

```

```
*** call the macro fileloop to loop through it all ***;  
%fileloop;  
  
/*  
proc datasets lib=work memtype=data kill; run; quit;  
*/
```