Welcome to the Michigan SAS Users Group (MSUG) Conference at MMS Holdings, Inc.

MMS Holdings, Inc.

MEDICAL WRITING

CLINICAL PROGRAMMING

BIOSTATISTICS

DATA MANAGEMENT

REGULATORY AFFAIRS

MEDICAL INFORMATION

DRUG SAFETY

CLINICAL DEVELOPMENT SUPPORT

GMP AND GLP SERVICES

Utilizing SAS for Efficient Coding

Michelle Gayari

MSUG

November 5, 2009

MMS Holdings Inc.
6880 Commerce Blvd.
Canton, MI. 48187
Phone: 734.245.0310
Fax: 734.245.0320
www.mmsholdings.com

ISO 9001:2008

MMS Holdings Inc.

Environment

- Program faster, more efficient, with fewer errors
- Always multiple ways to program task
- Utilize SAS to create datasets, variable names, etc to make code more efficient
- Able to reuse code without re-validation
- Create more output with single program
- Simple example

Problem

- Create a unique subject id in each dataset consisting of combination of study number, site and subject number
  - 38 studies
- Each study has different
  - Dataset names
  - Number of datasets per study
  - Variable names for subject and site number

Multiple dataset names

Ex: study 205

1	AE
2	ANXHIST
3	ANXIETY
4	CONCMDRG
5	DEMO
6	DOSE
7	FINAL
8	GLOBAL
9	HAMILTON
10	LABDATA

Ex: study 555

1	AE_EVENT
2	ANXHIST
3	DEMO
4	EDAM
5	EVAL_ULCER
7	HAMILTON
8	PEN
9	PSYCHIATRIC

Ex: study 550

1	DEMO
2	HAM
3	GLOBAL

Solution

- Create macro to run for each study
- One possibility – list unique dataset names for each study, for example:

1	AE
2	ANXHIST
3	ANXIETY
4	CONCMDRG
5	DEMO
6	DOSE
7	FINAL
8	GLOBAL
9	HAMILTON
10	LABDATA

```
%let &ds=AE;
```

```
data &ds;
  set &ds;
  usubjid=put(&study,z4.)||'_'||put(&site,z4.)||'_'||put(&id,z4.);
run;
```

```
%macro addstudy (ds=ds, study=study, site=site, id=id);
```

MMS Holdings Inc.

Better Solution

- Let ***proc contents*** do the work for you

```
proc contents data=&lib.._all_ out=test;
run;

proc sql;
   create table test1 as
   select distinct memname
   from test;
quit;
```

Better Solution

- **This creates dataset *TEST1* with all dataset names**

Obs	MEMNAME
1	AE
2	ANXHIST
3	ANXIETY
4	CONCMDRG
5	DEMO
6	DOSE
7	FINAL
8	GLOBAL
9	HAMILTON
10	LABDATA

Count number datasets

- Next loop through each dataset name to create USUBJID
- Say one study has 10 datasets and another has 12 or 5
- &obs could be additional macro variable to be entered for each study,

  i.e. %let %obs=10; then

  %do i=1 %to &obs;

Multiple dataset names

Ex: study 205

1	AE
2	ANXHIST
3	ANXIETY
4	CONCMDRG
5	DEMO
6	DOSE
7	FINAL
8	GLOBAL
9	HAMILTON
10	LABDATA

Ex: study 555

1	AE_EVENT
2	ANXHIST
3	DEMO
4	EDAM
5	EVAL_ULCER
7	HAMILTON
8	PEN
9	PSYCHIATRIC

Ex: study 550

1	DEMO
2	HAM
3	GLOBAL

Count number datasets

- **Create macro variable %obs from _n_**

```
data test2;
    set test1;
    obs=_n_;
run;

proc sort data=test2; by descending obs;
proc sql;
    select  obs into:obs
    from test2;
quit;
```

Transpose

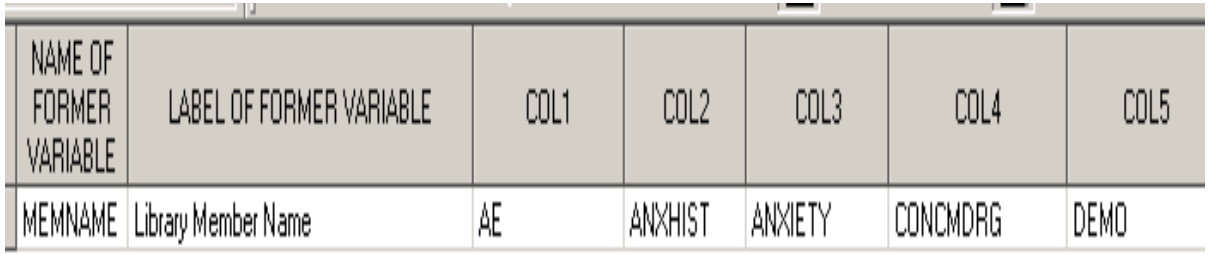
```
proc transpose data=test2 out=test3;
   var memname;
run;
```

NAME OF FORMER VARIABLE	LABEL OF FORMER VARIABLE	COL1	COL2	COL3	COL4	COL5
MEMNAME	Library Member Name	AE	ANXHIST	ANXIETY	CONCMDRG	DEMO

Loop

```
%do i=1 %to &obs;

proc sql;
    select col&i into:ds&i
    from test3;
quit;
```



NAME OF FORMER VARIABLE	LABEL OF FORMER VARIABLE	COL1	COL2	COL3	COL4	COL5
MEMNAME	Library Member Name	AE	ANXHIST	ANXIETY	CONCMDRG	DEMO

- MLOGIC(ADDSTUDY): %DO loop beginning; index variable I; start value is 1; stop value is 10; by value is 1.
- **MPRINT(ADDSTUDY): proc sql;**
- SYMBOLGEN: Macro variable I resolves to 1
- SYMBOLGEN: Macro variable I resolves to 1
- **MPRINT(ADDSTUDY): select col1 into:ds1 from test3;**
- **MPRINT(ADDSTUDY): quit;**

Loop

- Loop continued:

```
data interim.&&ds&i;
    set in.&&ds&i;
    USUBJID=put(&study,z4.)||'-'||put(&site,z4.)||'-'||put(&id,z4.);
run;
%end;
```

Log

- SYMBOLGEN: && resolves to &.
- SYMBOLGEN: Macro variable I resolves to 1
- SYMBOLGEN: Macro variable DS1 resolves to AE
- **MPRINT(ADDSTUDY): data interim.AE ;**
- SYMBOLGEN: && resolves to &.
- SYMBOLGEN: Macro variable I resolves to 1
- SYMBOLGEN: Macro variable DS1 resolves to AE
- **MPRINT(ADDSTUDY): set in.AE ;**
- SYMBOLGEN: Macro variable STUDY resolves to 205
- SYMBOLGEN: Macro variable SITE resolves to 0001
- SYMBOLGEN: Macro variable ID resolves to subject
- **MPRINT(ADDSTUDY): USUBJID=put(205,z4.)||'-'||put(site,z4.)||'-'||put(subject,z4.);**
- **MPRINT(ADDSTUDY): run;**

- NOTE: There were 87 observations read from the data set IN.AE.
- NOTE: The data set INTERIM.AE has 87 observations and 17 variables.

MMS Holdings Inc.

Conclusions

- One macro was able to be run
  - across 38 studies,
  - with multiple dataset names,
  - different number of datasets per study,
  - and different variable names for subject and site numbers

MMS Holdings Inc.

Conclusions

- Only inputs needed
  - &study = study number
  - &id = variable name for subject number
  - &site = variable name for site

```
%macro addstudy (study=study, site=site, id=id);
```

MMS Holdings Inc.

Conclusions

- With flexibility of macro you are able to do more work with minimal changes
- Let SAS create datasets and variables based on the data so you don’t have to
- Less chance for errors, rework and revalidation