

The Evolution of a Macro

Converting your SAS® files to a Different Version

John Charles Gober, Deborah Mullen, Jana Smith
Bureau of the Census, Demographic Surveys Methodology Division

INTRODUCTION

The purpose of this paper is not to demonstrate the best way to convert SAS data files from version six to version eight but to show the process by which a macro was created and evolved to meet the needs of our users. The macro was designed by several people who worked independently and then came together to discuss their different approaches in order to optimize the macro for the majority of our users.

THE TARGET USER

The Bureau of the Census has users ranging from interns to those who have been programming using the SAS System for over twenty years. Due to the large volume and variety of data that we receive and process for our sponsors it is important that our code be efficient, flexible, yet highly maintainable by all levels of these programmers. That is why we not only have formal classes in SAS, but also have several Census SAS users groups, toolboxes of SAS snippets, and formal and informal code reviews, etc. The programmers new to either SAS or our projects get brought up to speed in a relatively short time and just as important the more experienced programmers get to learn the up and coming tricks of the newbie's.

CONVERSION IS SIMPLE...

If you just need to convert an entire library the process is simple. Just create two LIBNAME statements and then use a simple PROC COPY.

```
libname myv6 v6 'c:\mylib';  
libname myv8 v8 'c:\mylib';  
  
proc copy in=myv6 out=myv8;  
run;
```

However, PROC COPY would not remove the old version six datasets from the associated directory path (an incorrect assumption from one of our veterans, co-author number one) so special code should be written to accommodate this feature. Hence, the start of the macro.

MACRO VERSION ONE

In our first macro we wanted to give the user an option to convert and save the version six datasets or to remove them entirely from the directory. So the following code was created:

```
%macro convert(dirname=, del=n);  
  libname in1 v6 "&dirname";  
  libname out1 v8 "&dirname";  
  
  *create macro variable of v6 datasets;  
  proc sql noprint;  
    select distinct memname  
    into : memlist separated by ' '  
    from sashelp.vmember  
    where engine='V6' & libname='IN1' &  
          memtype='DATA';  
  quit;  
  
  proc copy in=in1 out=out1  
            memtype=data;  
  run;  
  
  %if %lowercase(&del) = yes %then  
  %do;  
    proc datasets lib=in1  
                  memtype=data;  
      delete &memlist;  
    run;  
  %end;  
%mend convert;  
%convert (dirname=c:\mylib,del=yes)
```

MACRO VERSION TWO

After realizing that PROC COPY can delete (or more accurately MOVE) datasets it was decided to eliminate the PROC SQL code and redesign the macro. It was also at this point that it was decided to give the user an additional option of copying and converting the datasets to a new directory thus maintaining the integrity of the existing version six library. Thus the following code was created.

```
%macro convert(v6lib=,v8lib=,  
              del=no);  
  libname in1 v6 "&v6lib";  
  libname out1 v8 "&v8lib";
```

```

proc copy in=in1 out=out1
    memtype=data
    %if %lowercase(&del) = yes %then
    %do;
        move;
    %end;;
    *note the double semi colon above;
    run;
%mend convert;

%convert (v6lib=c:\sesug2003,
        v8lib=c:\sesug2003\v8,
        del=no)

```

MACRO VERSION THREE

At this point in time one will notice that this macro is starting to look extremely similar to the complete PROC COPY procedure offered by SAS. So why continue developing this macro? As mentioned before we have users of all skill levels working for the Bureau. Some may have never been exposed to all of the intricacies of PROC COPY. Others may not have time to read up on the pre-designed procedures.

There are also users who do not use Base SAS® but find a rare circumstance where they need to undertake a dataset or library migration. These could be our SAS/Graph®, SAS/IML®, or our SAS/IntrNet® users. Users who do not need to know the full power of PROC COPY but just need something special to get them by.

The final two options that were incorporated into the macro were the ability to convert only selective datasets, useful when your library contain datasets from multiple projects or the need to turn over final versions of datasets to an outside user, and also the ability to transfer over indexes. The other options of PROC COPY could also have been added but at this point it was decided that the macro was balanced with just enough options to cover the most common circumstances without becoming too cumbersome.

```

%macro convert(v6lib=,
              v8lib=,
              list=,
              index=no,
              del=no);
    libname in1 v6 "&v6lib";
    libname out1 v8 "&v8lib";
    proc copy in=in1 out=out1
        memtype=data
        %if %lowercase(&del) = yes %then
        %do;
            move;

```

```

    %end;
    %if %lowercase(&index) = yes %then
    %do;
        index=yes;
    %end;;
    *note the double semi colon above;
    %if %length(&list) > 0 %then
    %do;
        select &list;
    %end;
    run;
%mend convert;
%convert (v6lib=c:\sesug2003,
        v8lib=c:\sesug2003\v8,
        del=no,
        list=temp1 temp2
            stfile01-stfile20
            cty:,
            index=yes)

```

Note the variety of ways to list the files to be converted.

CONCLUSION

In Version 9 SAS has created PROC MIGRATE to help with the conversion process of older SAS Datasets. So why create this macro? It was created as an exercise in thought. How to take a simple idea or application and make it useful without overwhelming the user. Macros can be great tools and timesavers but only if they are kept manageable to the average users of the environment.

CREDITS AND ACKNOWLEDGEMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Oracle is a registered trademark, and ConText, Oracle Alliance and Oracle8 are trademarks or registered trademarks of Oracle Corporation.

CONTACT INFORMATION

Questions and comments concerning this paper can be directed to:

john.charles.gober@census.gov
 deborah.m.mullen@census.gov
 jana.l.smith@census.gov
 Bureau of the Census
 FOB3 MS8700
 Washington DC 20233-8700