Paper CC58

PROC CATALOG, the Wish Book SAS® Procedure

Louise Hadden, Abt Associates Inc., Cambridge, MA

ABSTRACT

SAS® data sets have PROC DATASETS, and SAS catalogs have PROC CATALOG. Find out what the little known PROC CATALOG can do for you!

INTRODUCTION

Michael Raithel characterizes PROC DATASETS as the "Swiss Army Knife of SAS Procedures" (Raithel, 2011). PROC DATASETS can do an amazing array of tasks relating to SAS data sets; PROC CATALOG is a similar, utilitarian procedure. It is handy (like a Leatherman® tool!) itself, and in conjunction with other SAS procedures can be very helpful in managing the special SAS files that are SAS catalogs.

WHAT ARE SAS CATALOGS?

SAS catalogs are specialized SAS files which may hold many entries or elements. Common types of SAS catalogs include format catalogs or libraries, macro catalogs, graphics catalogs, and template catalogs. You can also catalog LOG output, LISTING output and SOURCE files (programs.) SAS catalogs can be used to catalog and manage SAS processing.

A SAS catalog may include one or more entries. The default name and extension for a format catalog is formats.sas7bcat, but you don't have to, and shouldn't, resort to the default for user-defined formats. Macro, graphic and template catalogs can be named and described in the processes that create them; and also have the extension sas7bcat, for example, sasmacr.sas7bcat in the SASHELP folder. Names and descriptions can greatly enhance ease and reproducibility in data processing – and PROC CATALOG helps get us there. Different types of entries can exist in the same catalog.

HOW DO YOU CREATE SAS CATALOGS?

To save a format or formats in a permanent format catalog, use PROC FORMAT to create a format by either defining value statements or by using PROC FORMAT's CNTLIN statement. Note the use of a specific format catalog name instead of the default formats.sas7bcat, which makes it easier to identify the specific formats needed for a task. In addition, if you are creating several different types of catalog entries for a project, make sure to distinguish between format, macro, and template catalogs either by name or by locating them in separate folders to avoid overwriting your entries.

```
PROC FORMAT LIBRARY = LIBRARY.Compendium2011Res;
    VALUE YESNODK 1 = "Yes"
                  2 = "No"
                  8 = "Don't Know";
RUN;

PROC FORMAT LIBRARY = LIBRARY.Compendium2011Res CNTLIN=cntl_Antipsych;
RUN;
```

Macros can be stored in macro catalogs when they are defined. Again, using a specific catalog name can be helpful. Note that while creating and storing a macro, you can provide a helpful description to document your macro.

```
OPTIONS MSTORED SASMSTORE=MYSTORE;

LIBNAME MYSTORE "S:\PROJECTS\SGF2015\COMPILED_MACROS";

%MACRO RUNCONTS (IN=, OUT=) / STORE SOURCE DES="Run Contents and Test Print";

    TITLE1 "CONTENTS AND TEST PRINT OF DS &IN. - &SYSDATE - &SYSTIME";
    PROC CONTENTS DATA = &IN. VARNUM;
    RUN;

    PROC PRINT DATA= &IN. (OBS=5) NOOBS;
    RUN;

%MEND;
```

Graphics catalogs can be created in SAS/GRAPH using the GOUT statement.   If you do not specify a permanent SAS catalog with a library and catalog name, SAS/GRAPH uses WORK.GSEG to store graphic output. Note that you can provide a description while creating your graphic image; however, in this case, the map was destined to be "replayed" using PROC GREPLAY into a page with another map, so the description is left blank.

```
/* Create two graphs to replay on a single page and store in a graphics catalog
   */

PROC GMAP DATA=my_data MAP=mymap ALL ANNO=extra_label GOUT=library.excat;
ID state_name;
CHORO &invar._5 / DISCRETE COUTLINE = gold WOUTLINE=1
CDEFAULT=yellow
LEGEND=legend1
HTML=my_html
DES='' name="&y2.";
RUN;

QUIT;
```

Graphics templates can be stored in SAS catalogs as well. Below follows an example of a replayed graphic using both maps from a catalog and a template stored in a catalog.

```
/* Create a template with one up, one down pattern with space in between to
   replay two maps in */
PROC GREPLAY TC=LIBRARY.TEMPCAT NOFS;
   TDEF V2S des= "1 BOX UP, 1 BOX DOWN (WITH SPACE)"
        1/llx=0    lly=52
          ulx=0    uly=100
          urx=100  ury=100
          lrx=100  lry=52

        2/llx=00  lly=00
          ulx=00  uly=48
          urx=100 ury=48
          lrx=100 lry=0;
QUIT;

/* Replay the graphs with a template to create one graph. The graphs stored in
   LIBRARY.EXCAT are replayed to create one graph. */
```
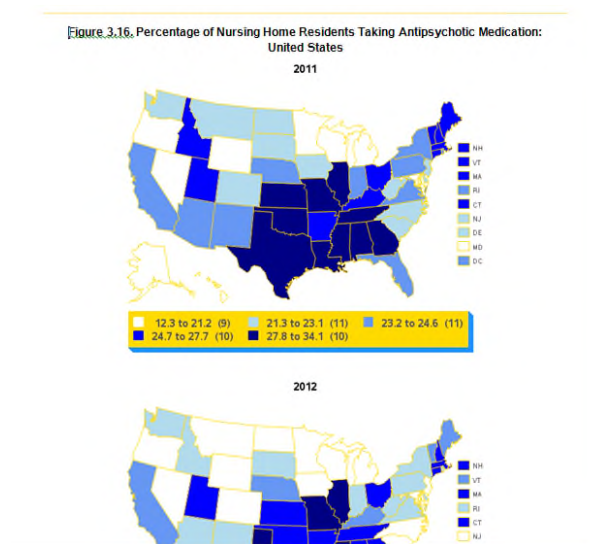
```
PROC REPLAY IGOUT=library.excat NOFS
   TC=library.tempcat TEMPLATE=v2s;
   TREPLAY 1:&y2._11 2:&y2._12 NAME="&y2._c";
QUIT;
```


Figure 3.16. Percentage of Nursing Home Residents Taking Antipsychotic Medication: United States

SO WHAT CAN I DO WITH PROC CATALOG? REAL LIFE APPLICATIONS

PROC CATALOG provides us with a way to describe and manage SAS catalogs. You can create a listing of the contents of a catalog, copy an entire catalog or entries within a catalog, rename, swap and delete entries within a catalog, and enter or modify the description of a catalog entry. PROC CATALOG is an interactive procedure (but may be run in batch) and can be terminated by using a QUIT statement.

- To change the name of a catalog entry, use the **CHANGE** command.
- To list the contents of a catalog, use the **CONTENTS** command.
- To copy one or more entries in a catalog to another catalog, use the **COPY** command.
- To delete one or more entries in a catalog, use the **DELETE** command.
- To swap entry names in a catalog, use the **EXCHANGE** command.
- To prevent entries from being copied, use the **EXCLUDE** command.
- To enter or change the description of an entry, use the **MODIFY** command.
- To choose entries to copy, use the **SELECT** command.
- The misnamed **SAVE** command deletes all entries except those specified on the command.

The below sample shows how to modify a format by adding a description, and then lists the contents of the format catalog. Note that PROC CATALOG is the ONLY way to add a description to a format, as PROC FORMAT does not have a LABEL or DESCRIPTION option. The PROC CATALOG listing shows the entry name, type, dates and description.

```
PROC CATALOG CATALOG = LIBRARY.Compendium2011Res ;
MODIFY ANTIPSYCHF FORMAT (DESCRIPTION = "Legend labels for map [Antipsych]") ;
CONTENTS;
RUN ;
QUIT;
```

```
Contents of Catalog LIBRARY.COMPENDIUM2011RES

#  Name        Type    Create Date   Modified Date   Description
-------------------------------------------------------------------------------
1  ANTIPSYCHF  FORMAT  07/21/2014    07/21/2014    Legend labels for map [Antipsych]
```

.

For more detail on FORMAT type entries, use PROC FMTLIB. Note that PROC CATALOG provides a different set of information from PROC FORMAT FMTLIB, and that both outputs are useful.

```
PROC FORMAT LIBRARY=LIBRARY.Compendium2011Res FMTLIB;
RUN;
```

```
------------------------------------------------------------------------------
|                    FORMAT NAME: ANTIPSYCHF LENGTH: 18                       |
|    MIN LENGTH:  1  MAX LENGTH:  40  DEFAULT LENGTH:  18  FUZZ: STD          |
|----------------------------------------------------------------------------|
|START           |END             |LABEL  (VER. 9.4      24MAR2015:13:35:29)|
|----------------+----------------+------------------------------------------|
|               0|               0|12.3 to 20.3  (10)                         |
|               1|               1|20.4 to 21.7  (10)                         |
|               2|               2|21.8 to 23.3  (11)                         |
|               3|               3|23.4 to 26.7  (10)                         |
|               4|               4|26.8 to 33.1  (10)                         |
------------------------------------------------------------------------------
```

PROC CATALOG listings for other types of entries are similar, as seen in the examples below. Note this catalog (provided in the SASHELP directory by SAS) does not have descriptions.

```
PROC CATALOG CATALOG = SASHELP.SASMACR ;
CONTENTS;
RUN ;
QUIT;
```

```
Contents of Catalog SASHELP.SASMACR

#   Name                     Type    Level        Create Date        Modified Date      Description
--------------------------------------------------------------------------------------------------------
1   AARFM_EXEC               MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
2   EM_RFM_CONTROL           MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
3   EM_RFM_INDEPENDENT_RANK  MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
4   EM_RFM_NESTED_RANK       MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
5   EM_RFM_SET_SCORE         MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
6   EM_RFM_TRAN_2_CUST       MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
7   TS_CREATE_DSN            MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
8   TS_CREATE_DSRC           MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
9   TS_DROP_DSN              MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
10  TS_DROP_DSRC             MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
11  TS_PW_QUOTE              MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
12  TS_UPDATE_SCKEY          MACRO   2      12/05/2013 01:16:26   12/05/2013 01:16:26
```

To add a description to a macro entry, we first copy the SASHELP macro catalog to a temporary catalog, then modify one of the entries.

```
PROC CATALOG CAT=SASHELP.SASMACR;
   COPY OUT=TEMPCAT;
RUN;

PROC CATALOG CAT=TEMPCAT;
   MODIFY TS_CREATE_DSN.MACRO (DES = 'Macro to Create DSN');
   CONTENTS;
```

```
RUN;
QUIT;
```

```
Contents of Catalog WORK.TEMPCAT

 #    Name                      Type      Create Date         Modified Date       Description
-----------------------------------------------------------------------------------------------------
 1    AARFM_EXEC                MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 2    EM_RFM_CONTROL            MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 3    EM_RFM_INDEPENDENT_RANK   MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 4    EM_RFM_NESTED_RANK        MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 5    EM_RFM_SET_SCORE          MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 6    EM_RFM_TRAN_2_CUST        MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 7    TS_CREATE_DSN             MACRO     12/05/2013 01:16:26  03/24/2015 11:27:41  Macro to Create DSN
 8    TS_CREATE_DSRC            MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
 9    TS_DROP_DSN               MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
10    TS_DROP_DSRC              MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
11    TS_PW_QUOTE               MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
12    TS_UPDATE_SCKEY           MACRO     12/05/2013 01:16:26  12/05/2013 01:16:26
```

CONCLUSION

While PROC CATALOG is not as familiar as the workhorse procedures such as PROC DATASETS, it provides another handy tool for SAS programmers to manage entries in their SAS catalogs. In conjunction with complementary procedures and data steps, PROC CATALOG can greatly enhance the ability to document and manage formats, macros, graphics catalogs, template catalogs, SAS output catalogs (logs, lists) and SAS source code catalogs.

REFERENCES

Gilbert, Jeffrey D., 2000. "Replaying Graphics with PROC GREPLAY." *Proceedings of SUGI 25.*

Chapman, David D., 2004. "Using SAS® Catalogs to Develop and Manage SAS® DATA Step Programs." *Proceedings of SUGI 29.*

Li, Stan, 2011. "Some Useful Techniques of PROC FORMAT." *Proceedings of PharmaSUG 2011.*

Madduri, Kavitha, 2011. "Macros to Help You Clean Up!" *Proceedings of PharmaSUG 2011.*

Raithel, Michael A., 2011. "PROC DATASETS: the Swiss Army Knife of SAS® Procedures." *Proceedings of SAS Global Forum 2011.*

Rosenbloom, Mary F.O. and Lafler, Kirk Paul, 2012. "Best Practices: Clean House to Avoid Hangovers." *Proceedings of SAS Global Forum 2012.*

Russell, Rob, 2010. "Using Data Set Labels and Format Descriptions to Simplify Application Debugging." *Proceedings of SAS Global Forum 2010.*

Stojanovic, Mirjana and Hollis, Donna, 2005. "Ways to Store Macro Source Codes and How to Retrieve Them." *Proceedings of SESUG 2005.*

Shoemaker, Jack, 1998. "Creating a Self-Documenting FORMAT catalog." *Proceedings of NESUG 98.*

Thornton, Patrick, 2008. "Documenting SAS® Macro Programs using CATALOGS." *Proceedings of WUSS 2008.* Available at http://sascommunity.org/wiki/Documenting_SAS_Macro_Programs_using_CATALOGS

Williams, Christianna. 2014. "FORMATs Top Ten" *Proceedings of SESUG 2014.*

Williams, Christianna. 2014. "PROC RANK, PROC SUMMARY, and PROC FORMAT Team Up and a Legend is Born!" *Proceedings of SESUG 2014.* Available at http://analytics.ncsu.edu/sesug/2014/AD-07.pdf

Wright, Philip A., 2010. "Eliminating Redundant Custom Formats (or How to Really Take Advantage of PROC SQL, PROC CATALOG, and the DATA step)." *Proceedings of SAS Global Forum 2010.*

ACKNOWLEDGMENTS

CONTACT INFORMATION

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

Name: Louise Hadden
E-mail: Louise_Hadden@abtassoc.com

Look for this and other papers on www.sascommunity.org for tip sheets, paper copies, presentations and more.