

## Ms. Independence (from the SAS® Format Library)

Nancy McGarry and Louise Hadden, Abt Associates Inc.

### ABSTRACT

SAS® practitioners are frequently called up to format variables in SAS datasets they have received or created for various use cases. Analysts and other end users desire the convenient categorization, transformative nature, and attractive appearance that SAS formats can lend to variables for reports and further analytic and data set construction purposes. SAS formats can be created in SAS work space, and can be stored permanently in SAS catalogs, a specially purposed container for SAS files. SAS formats created in work space are ephemeral, and only exist for the duration of a SAS session. SAS formats stored in a SAS catalog are notoriously difficult to transfer across platforms, SAS versions and “bit” versions (32 bit vs 64 bit). Recipients of SAS data sets with “embedded formats” and/or SAS catalogs originating from incompatible systems find themselves in a quandary – SAS reports errors when it can’t find a compatible catalog (IF a catalog accompanies a data set) for formats permanently associated with variables in a SAS data set. SAS catalogs are also very difficult to update, document and manipulate. This paper and presentation / poster propose straightforward SAS solutions for the creation, transfer and use of SAS formats.

### INTRODUCTION

The bane of every SAS programmer’s existence are mystery data sets: those data sets with unknown ancestry and little to no information regarding the contents of the data set, delivered without the comfort of well-written documentation and background information such as a survey instrument. Most of us, when faced with such a “mystery box”, resort to PROC CONTENTS, a test print, maybe some freqs and means. If you are lucky, you gain useful information from this strategy. If you are REALLY lucky, you can proceed with your analytic and/or file building tasks. All too often, it’s a worst case scenario: you may not have noticed the presence of user-defined formats in the PROC CONTENTS output in your testing, and/or you went straight to a test print or frequencies. The next thing you know you see the dreaded error message in your log:

```
ERROR: The format $YNDKF. was not found or could not be loaded.
```

Reality comes crashing in. Sometimes SAS data sets are delivered with embedded formats, and the expected format catalog is either not supplied, is incompatible due to system/platform/bit differences, is outdated, is corrupted, and/or is incomplete.

```
OPTIONS nofmterr;
```

This little SAS system option is a quick bandaid that will allow you to open and process a data set with embedded formats that you do not have a catalog for, but you will not have the benefit of the formats or view the data as it was intended to be.

```
OPTIONS FMTSEARCH=(lib.survey1 lib.survey2 lib2.survey3);
```

At other times, the problem may be the location of the missing library. Using the fmtsearch option allows you to load format catalogs in the order that they are mentioned. Note that the librefs are tied to the libname statements in your program.

But what do you do if you still don’t have a format catalog for your data set?

## SOLUTIONS FOR FORMAT CATALOGS

### FOR DATA SETS WITH EMBEDDED FORMATS:

If the data set that has been delivered without being properly attired is from a known source that has been contracted to provide the data, request that incoming data sets be delivered without embedded formats and with well documented programs to (a) create a collection of formats and (b) associate those formats with variables in the delivered data set(s).

If you are unable to obtain better documentation and/or your format catalog, not all is lost. Strip embedded formats with the option NOFMterr described above and a data step by simply reformatting variables with an “empty” format statement. PROC DATASETS , ATTRIB statement or PROC SQL also work well to (a) create a collection of formats and (b) associate those formats with variables in the delivered data set(s). Documentation of the data set(s) certainly helps in this effort.

```
DATA WANT;  
  SET HAVE;  
  FORMAT _ALL_;  
RUN;
```

Strip embedded formats and (re) build formats and format associations using techniques described in detail below.

## NOW YOU HAVE ASSOCIATED FORMATS IN YOUR FILE, WHAT DO YOU DO?

### TO CREATE TRANSPORTABLE, MODIFIABLE FORMATS TO ASSOCIATE WITH VARIABLES IN DATA SETS OR IN PROCEDURES:

Build a SAS data set to house elements of a format catalog that can be transformed into catalogs via PROC FORMAT CNTLIN

```
PROC FORMAT LIBRARY = LIBRARY.CHILD_DAY CNTLIN=PRVTYPID;  
RUN;
```

Build a SAS data set from an existing SAS format catalog via PROC FORMAT CNTLOUT that can be used for PROC FORMAT CNTLIN. Just as you can use PROC FORMAT CNTLIN to read a SAS data set into a temporary or permanent format catalog, you can also get information OUT of a format catalog. A SAS data set is generated with an inordinate number of columns, of which only five or six are commonly used, FMTNAME, TYPE, START, END, LABEL and HLO (only three are required – FMTNAME, START and LABEL).

```
PROC FORMAT = LIBRARY.CHILD_DAY;  
  CNTLOUT = CNTLOUT;  
RUN ;  
QUIT;
```

WYO – write your own format!

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

Additionally, you can use an include statement that contains PROC FORMAT statements to read in formats within your SAS program, and then apply them to a work catalog so that they can be used. If you can locate electronic survey instrument or existing documentation, it's possible to process these files to create a format creation program.

## RESEARCH AND DOCUMENT YOUR FORMAT FILES

Whether or not you have been supplied with a working format catalog, it's worthwhile to "own" your format files. Gaining familiarity with your data and formats and how they interact can be extremely informative, allowing you to see what is actually going on with data. In the processing of testing your efforts to provide formats for your data set(s), quality assurance can be achieved, for example, tabulating formatted and unformatted data. This increased knowledge of the data will enhance your standing with other stakeholders.

### SAS format catalogs

Use identifiable two part format catalog names i.e. survey1.sas7bcat, survey2.sas7bcat instead of formats.sas7bcat. 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.

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.CHILD_DAY;
MODIFY PRVTYPID FORMAT (DESCRIPTION = "Categories for Provider Type ID") ;
CONTENTS;
RUN ;
QUIT;
```

Contents of Catalog LIBRARY.CHILD\_DAY

#	Name	Type	Create Date	Modified Date	Description
1	PRVTYPID	FORMAT	07/21/2019	07/21/2019	Categories for Provider Type ID

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.CHILD_DAY FMTLIB;
RUN;
```

FORMAT NAME: PRVTYPID LENGTH: 24			
MIN LENGTH: 1	MAX LENGTH: 40	DEFAULT LENGTH: 24	FUZZ: STD
START	END	LABEL (VER. 9.4 24MAR2019:13:35:29)	
CCC		CCC	Child Care Center
HS		HS	Healthy Start
AR		AR	At Risk
OSH		OSH	Outside of School Hours

## SAS data sets containing formats

```

* 1: map provnum to fips county code ;
data ctrl_prov_fips;
  length label $ 3 BASEVAR $ 32;
  set prov_xwalk2 (keep = provnum county_fips
                  rename=(provnum=start county_fips=label)) end=last;
  retain fmtname '$FIPSCTY' type 'c' BASEVAR 'PROVNUM';
  output;
  if last then do;
    hlo='O';
    label='***';
    output;
  end;

keep fmtname type start label hlo basevar;
run;

proc sort data = ctrl_prov_fips ;
by fmtname start;
run;

. . .

* Put the formats together, setting length long enough to accommodate longest
label;

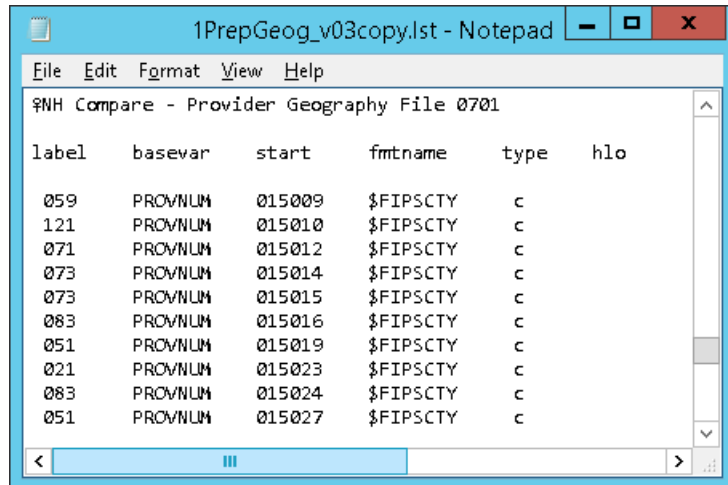
data allgeogformats_&fileyear.&filedate. ;
length fmtname $8 type $1 start $14 label $255;
set ctrl_prov_fips
  ctrl_prov_cntynm
  ctrl_prov_cbsa
  ctrl_prov_urban
  ctrl_prov_regionc
  ctrl_prov_fips_st
  ;
by fmtname start ;

label fmtname = 'Name of Format'
      start = 'Start of Range for Format'
      label = 'Label for Format'
      type = 'Type of Format'
      ;

run;

```

```
* test building the formats ;
proc format library=work cntlin= allgeogformats_&fileyear.&filedate. ;
run;
```



label	basevar	start	fmtname	type	hlo
059	PROVNUM	015009	\$FIP SCTY	c	
121	PROVNUM	015010	\$FIP SCTY	c	
071	PROVNUM	015012	\$FIP SCTY	c	
073	PROVNUM	015014	\$FIP SCTY	c	
073	PROVNUM	015015	\$FIP SCTY	c	
083	PROVNUM	015016	\$FIP SCTY	c	
051	PROVNUM	015019	\$FIP SCTY	c	
021	PROVNUM	015023	\$FIP SCTY	c	
083	PROVNUM	015024	\$FIP SCTY	c	
051	PROVNUM	015027	\$FIP SCTY	c	

You can then use this data set for reporting and/or to create a format statement.

## CONCLUSION

The inability to link a format catalog with its data is a frequent quandary for SAS programmers. We hope that you are able to use some of the ideas we have presented here in the future.

## REFERENCES

- Boniface, Christopher. 2017. "Instant Formats in a Blink with PROC FORMAT CNTLIN=" *Proceedings of SAS Global Forum*. Available at <https://support.sas.com/resources/papers/proceedings17/0971-2017.pdf>
- Hadden, Louise. 2015. "PROC CATALOG, the Wish Book SAS® Procedure" *Proceedings of SESUG 2015*. Available at <https://analytics.ncsu.edu/sesug/2015/CC-58.pdf>
- Hughes, Troy Martin. 2018. "Abstracting and Automating Hierarchical Data Models: Leveraging the SAS® FORMAT Procedure CNTLIN Option To Build Dynamic Formats That Clean, Convert, and Categorize Data" *Proceedings of SESUG 2018*. Available at [https://analytics.ncsu.edu/sesug/2018/SESUG2018\\_Paper-224\\_Final\\_PDF.pdf](https://analytics.ncsu.edu/sesug/2018/SESUG2018_Paper-224_Final_PDF.pdf)
- Shoemaker, Jack. 2017. "Ten Things You Should Know About PROC FORMAT" *Proceedings of WUSS 2017*. Available at [https://www.lexjansen.com/wuss/2017/209\\_Final\\_Paper\\_PDF.pdf](https://www.lexjansen.com/wuss/2017/209_Final_Paper_PDF.pdf)
- 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>

## CONTACT INFORMATION

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

Nancy McGarry  
Abt Associates Inc  
[Nancy\\_McGarry@abtassoc.com](mailto:Nancy_McGarry@abtassoc.com)

Louise S. Hadden  
Abt Associates Inc.  
[Louise\\_Hadden@abtassoc.com](mailto:Louise_Hadden@abtassoc.com)

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.

Other brand and product names are trademarks of their respective companies.