Paper 110-28

Using SAS Catalogs to Develop and Manage SAS DATA Step Program

David D. Chapman, US Census Bureau, Washington, DC

ABSTRACT

A key concept in program management is to develop the code in one place, make changes at one location, move the code to where it is needed for testing or use, and when making changes make sure they work before changing production code. SAS catalogs are ideal for developing, testing, and managing data step programs. This paper reviews SAS catalogs and the different kinds of elements they contain. It discusses Source, LOG, OUTPUT, Macro, Format, and CATAMS elements. For each it gives an example of how to put it into a catalog and then how to retrieve if from the catalog and use it. It also discusses how to manage and sequence code coming out of the catalog to make up your data step and procedure programs. The use of PROC CATALOG to view catalog contents and copy elements to another location, and PROC UPLOAD and PROC DOWNLOAD to move the SAS catalogs to a production location on a different computer are discussed.

SAS CATALOGS

SAS catalogs are special SAS files that store many different kinds of information in smaller units called catalog entries. Each entry has an entry type that identifies its purpose to the SAS System. A single SAS catalog can contain several different types of catalog entries. Some catalog entries contain system information such as key definitions. Other catalog entries contain application information such as window definitions, help windows, formats, informats, macros, or graphics output. You can list the contents of a catalog using various SAS features, such as SAS Explorer and PROC CATALOG.

SAS catalog entries are fully identified by a four-level name of the form: “libref.catalog.entry-name.entry-type”. An example is “SUGI28.PAPER110.CONTROL.SOURCE”. You commonly specify the two level name for an entire catalog, as follows: *libref.catalog (e.g. SUGI28.PAPER110)*. “libref” is the logical name of the SAS data library to which the catalog belongs. Catalog is a valid SAS name for the file.

The entry name and entry type are required by some SAS procedures. If the entry type has been specified elsewhere or if it can be determined from context, you can use the entry name alone. To specify entry names and entry types, use this form: entry-name. entry-type (e.g. control.source). “Entry-name” is a valid SAS name for the catalog entry; “entry-type” is assigned by SAS when the entry is created.

CATALOG ELEMENTS

A SAS catalog is a special type of SAS file that contains elements. Elements can be several different types. All elements in a file can the same type of element or all elements can be a different type. Different types of elements are described below. A catalog can contain all the code and information necessary to execute a data step program.

There are many different types of catalog elements. Elements most commonly used with a data step program are listed below. The use of these elements is given in later sections.

Types of Catalog Elements	
Source	FORMAT
LOG	MACRO
OUTPUT	CATAMS

SOURCE

This probably the most common element in use. It is the equivalent to a SAS program (e.g. program1.sas) file or ASCCII test file. These elements are used to hold basic data step or procedure code. There can be a different element for each data step or proc or a data step can be divided between several different elements. It is an element similar to a text file used to hold SAS code, original macro statements, or other general purpose statment such as options.

Code is added to a catalog either interactively or in batch. When it is created in batch a data _null_ statement is used. Example code is given below.

```
filename wrtsrc catalog
“sugi28.paper220.read.source;
    data _null_ ;
       file wrtsrc ;
       put “proc print” ;
    run;
```

The LOG associated with inserting this code in the catalog “mycat” in the library “work” is given below.

```
1     filename wrtsce   catalog
               "sugi28.paper110.data.read.source";
2       data _null_;
3          file wrtsce;
4          put "proc print ; run";
5          run;

NOTE: The file WRTSCE is:
   Catalog Name=SUGI28.PAPER110.READ.SOURCE,
   Catalog Page Size=4096,
   Number of Catalog Pages=4,
   Created=17:26 Friday, December 6, 2002,
   Last Modified=17:26 Friday, December 6, 2002,
   File Name=C:\Temp\SAS Temporary
   Files\_TD2820\data.sas7bcat,
        Release Created=8.0202M0,Host
        Created=WIN_PRO

NOTE: 1 record was written to the file WRTSCE.
      The minimum record length was 16.
      The maximum record length was 16.
```

FORMAT

Formats are commonly used with data step programs. Formats are used to recode variables, as the basis for classifications that are part of tabulate totals using PROC FORMAT or PROC REPORT, or to display data in output. PROC FORMAT is used to create the formats and store them in the catalog. Formats can be created, stored in a catalog, and retrieved from the catalog for later use.

User defined format are created using PROC FORMAT. During creation they can also be stored in the catalog. To create a format and store it into a catalog named “paper110” in the library “sugi28", you can use the following code.

```
libname sugi28 “c:\”
proc format   library=sugi28.paper110 ;
value   region
              ‘1'=”NorthEast”
              ‘2'=”NorthEast”
              ‘3'=”Central”
              ‘4'=”Central”
              ‘5'=”South”
              ‘6'=”South”
              ‘7'=”South”
              ‘8' =”West”
              ‘9' =”West” ;
run;
```

This format that assign census region based on census division can then be made available from the catalog file “paper110” by setting an option. Once a format has been stored in a catalog, it can be accessed by specifying the system option FMTSEARCH (). If you move the catalog, you must either change the name of the libname or redefine the libname to reflect the catalog new location.

```
Options fmtsearch (sugi28.paper110);
```

SAS searches for formats in a specific order. It first seaches the catalog WORK.FORMATS, then the LIBRARY.FORMATS catalog, and finally the CODE.PGM_CODE catalog.

CATAMS
A CATAMS element is an element for holding data. This data can be retrieved from the catalog to create a SAS data set or view. In theory, the size of the data set created is limited only by the size of the catalog that can be created. In practice, data sets created from CATAMS elements are often used to hold program parameters. One application uses a CATAMS element to hold IP addresses of different computer systems.
The LOG file associated with creating a CATAMS element from the data sets SASHELP.COMPANY is given below.

```
367  filename data catalog
"sugi28.paper110.company.catams";
368
369  data _null_;
370  file data;
371  set  sashelp.company;
372      PUT LEVEL1 $
373          LEVEL2 $
374          LEVEL3 $
375          LEVEL4 $
376          LEVEL5 $
377          JOB1   $
378          DEPTHEAD $  ;
379  RUN;

NOTE: The file DATA is:
      Catalog
Name=SUGI28.PAPER110.COMPANY.CATAMS,
      Catalog Page Size=4096,
      Number of Catalog Pages=6,
      Created=18:28 Tuesday, December 10, 2002,
    Last Modified=19:04 Tuesday, December 10,
                                         2002,
    File Name=c:\paper110.sas7bcat,
    Release Created=8.0202M0,HostCreated=WIN_PRO

NOTE: 48 records were written to the file DATAX.
      The minimum record length was 57.
      The maximum record length was 85.

NOTE: There were 48 observations read from the
data set SASHELP.COMPANY.

NOTE: DATA statement used:
      real time           0.58 seconds
      cpu time            0.01 seconds
```

Once entered into a CATAMS element, the data can be extracted later to create a SAS data set or data view. An example is given below.

```
387  DATA  COMPANY_VIEW/VIEW=COMPANY_VIEW;
388      INFILE DATAX;
389      INPUT LEVEL1 $
390          LEVEL2 $
391          LEVEL3 $
392          LEVEL4 $
393          LEVEL5 $
394          JOB1   $
395          DEPTHEAD $;
396        PUT _INFILE_;
397        RUN;
NOTE: DATA STEP view saved on file
WORK.COMPANY_VIEW.

NOTE: A stored DATA STEP view cannot run under a
different operating system.

NOTE: DATA statement used:
      real time           0.13 seconds
      cpu time            0.01 seconds
398
399
400  PROC PRINT DATA=WORK.COMPANY_VIEW
(OBS=10);RUN;

NOTE: The infile DATAX is:
      Catalog
Name=SUGI28.PAPER110.COMPANY.CATAMS,
      Catalog Page Size=4096,
      Number of Catalog Pages=6,
      Created=18:28 Tuesday, December 10, 2002,
      Last Modified=19:04 Tuesday, December 10,
                                         2002,
      File Name=c:\paper110.sas7bcat,
      Release Created=8.0202M0,Host
      Created=WIN_PRO

International Ai TOKYO ADMIN CONTRACTS So Suumi MANAGER 1
International Ai TOKYO ADMIN CONTRACTS Steffen Graff ASSISTANT 2
International Ai TOKYO ADMIN FINANCE Karin Schmidt ACCOUNTANT 2
International Ai LONDON ADMIN PERSONNEL Anne Bauer MANAGER 1

o o o { lines deleted to save space}

International Ai NEW YORK TECHN. SERVICES MIS Claire Voyant
TRANSLATOR 2
International Ai NEW YORK TECHN. SERVICES MIS Roy Hobbs TECH.
CONS. 2

NOTE: 48 records were read from the infile

DATAX.
      The minimum record length was 57.
      The maximum record length was 85.
NOTE: View WORK.COMPANY_VIEW.VIEW used:
      real time           0.04 seconds
      cpu time            0.01 seconds
```

```
NOTE: There were 10 observations read from the
data set WORK.COMPANY_VIEW.
NOTE: PROCEDURE PRINT used:
      real time           0.06 seconds
      cpu time            0.00 seconds
```

CATAMS elements allow the programer to keep parameters with the catalog.

LOG AND OUTPUT
A normal part of SAS programs is the production of a LOG and OUTPUT file. This information can be stored in the catalog containing the code. This allows for a catalog to contain both the OUTPUT and the LOG file describing the program's operation and the original code that made up the program. Doing this makes the code almost self documenting. This is particularly useful when the code may vary each time it is use and a new catalog is created for each use. The code below illustrates how to use the PRINTTO procedure to store results from the OUTPUT and LOG windows in the program catalog.

```
PROC PRINTTO  NEW
                LOG  =SUGI28.PAPER110.PROGRAM.LOG
               PRINT=SUGI28.PAPER110.PROGRAM.OUTPUT;
               RUN;

PROC CATALOG  CATALOG=ABC.TEST;
    CONTENTS;
    RUN;
```

The code above puts the LOG statements associated with executing the PROC CATALOG statement into the element program of type LOG and puts the output of the CONTENTS procedures into the element program of type OUTPUT.

PROGRAM RELATE CATALOG ELEMENTS
Most catalog elements are related to storage of data step code or program output. Some SAS procedures make special use of the catalogs. Three SAS procedures that do this are PROC REPORT, SAS/GRAPH, and the SQL/QUERY Window. PROC REPORT uses a catalog to store table code. SQL/Query window uses the catalog to store already written queries. An example of storing and recalling a PROC REPORT table is given below.

```
491  proc report data=sashelp.company
   outrept=sugi28.paper110.report02
   nowd;
492  column level1 level2 N;
493  define level1 /group;
494  define level2 / group;
495  define n / analysis sum;
496  run;

NOTE: Definition stored in
SUGI28.PAPER110.REPORT02.
NOTE: There were 48 observations read from the
data set SASHELP.COMPANY.
NOTE: PROCEDURE REPORT used:
      real time           0.19 seconds
      cpu time            0.01 seconds
```

This code retrieves the code for the report and produces another copy of the table.

```
499  proc report
500       data=sashelp.company
   report=sugi28.paper110.report02
          nowd;run;
NOTE: There were 48 observations read from the
data set SASHELP.COMPANY.
NOTE: PROCEDURE REPORT used:
      real time           0.15 seconds
```

CONTROLLING PROGRAM FLOW
To use the catalog in our data step programs, we need to retrieve the program code from the catalog in a specific sequence. In practice, this can be done by creating a "master" or "control" source element that manages the flow. The master element would look like the code below. This code submits the code in other source elements (e.g. program_a, program_b, program_c, and program_d) in sequence. The contents of the output and log windows are stored in the catalog paper110.

```
filename   CONTROL   catalog
       "sugi28.paper110.master.source";
data _null_;
file master ;
PUT "LIBNAME SUGI28 "C:\"; "
               PUT "FILENAME CONTROL CATALOG
                       "SUGI28.PAPER110";"
PUT "PROC PRINTTO NEW
    LOG =SUGI28.PAPER110.PROGRAM.LOG
     OUTPUT=SUGI28.PAPER110.PROGRAM.OUTPUT;"
PUT "%INCLUDE CONTROL(PROGRAM_A)/SOURCE2;"
PUT "%INCLUDE CONTROL(PROGRAM_B)/SOURCE2;"
PUT "%INCLUDE CONTROL(PROGRAM_C)/SOURCE2;"
PUT "%INCLUDE CONTROL(PROGRAM_D)/SOURCE2;"
PUT "PROC PRINTTO;RUN;"
```

Programs can may be started either interactively or in batch. The key to this is either to use the options on the pull down menus to put code in the program editor or to construct a SAS program file to do it in batch. Code can be submitted interactively by going to the file pull down menu and picking "Open Object" Once the catalog and entry are selected, you can pick "Open in editor" to put all the code into the program editor where it can be submitted.

MANAGING THE CATALOG
Once program code is developed in a catalog, it often needs to be moved from a development directory or computer to a production directory on the production computer. There are several ways to do this. When the catalog is moved to a different directory on the same computer, PROC CATALOG or PROC DATASETS can be used. When the catalog is to be moved onto a different computer SAS/CONNECT with PROC UPLOAD or PROC DOWNLOAD can be used.

PROC CATALOG
PROC CATALOG is a procedure that is used to manage and move a catalog. It can also be used to show the contents of a SAS cataloog. Moving elements from a test directory to a production directory is illustrated below. The LOG showing the results of a copy is given below.

```
7    libname sugi28  "c:\";

NOTE: Libref SUGI28 was successfully assigned as
follows:
      Engine:        V8
      Physical Name: c:\
7  !                          run;

8    proc catalog cat= sugi28.paper110  ;
9         copy    out= work.code;
10       run;

NOTE: Copying entry ONE.SOURCE from catalog
SUGI28.PAPER110 to catalog WORK.CODE.
NOTE: Copying entry READ.SOURCE from catalog
SUGI28.PAPER110 to catalog WORK.CODE.
```

```
NOTE: Copying entry THREE.SOURCE from catalog
SUGI28.PAPER110 to catalog WORK.CODE.
```

PROC CATALOG can be also be used in to list all the elements in a catalog. The code below illustrates this.

```
proc catalog   cat=work.data;
               contents;
               run;
```

The output from this catalog operation is given below.

```
Contents of Catalog WORK.DATA

# Name   Type   Create Date           Modified Date
_______________________________________________________
1 ONE    SOURCE 06DEC2002:18:24:31   06DEC2002:18:24:31
2 READ   SOURCE 06DEC2002:17:26:07   06DEC2002:17:26:07
3 THREE  SOURCE 06DEC2002:18:24:56   06DEC2002:18:24:56
```

PROC UPLOAD/DOWNLOAD AND SAS/CONNECT

SAS/CONNECT can be used to transfer files between two different computers or to use files on two different computers at the same time. One computer is referred to as the local computer (often a PC); the other the remote computer (often a UNIX or OpenVMS Server). PROC UPLOAD is used to move files from the local to the remote computer. PROC DOWNLOAD is used to move files from the remote computer to the local computer. The code below is remotely submitted after a connection has been made with the remote computer.

```
PROC UPLOAD     INCAT=DEVELOPKMENT.PGM_CODE
               OUTCAT=PRODUCTION.PGM_CODE;
RUN;
```

The key is to use the "incat" and "outcat" options. PROC UPLOAD and PROC DOWNLO0AD, part of transfer services under SAS/CONNECT, is another way to move catalogs between different computer systems. PROC UPLOAD and PROC DOWNLOAD work similar to PROC CATALOG. This is also a situation when remote library services under SAS/CONNECT may be appropriate.

TESTING NEW CODE

During program development or for program maintenance, you often need to replace or modify of the code. To avoid problems it is wise to try the new code before actually making the change. SAS makes this easier with CATNAME statement. The CATNAME statment logical concatenates two catalogs together under on logical name called a "catref".

The code below illustrates this.

```
  CATNAME work.program_code
                         (developent.test_code
                        project.production_code);
  filename code  catalog    work.program_code ;
  %include code (run) / source2;
  run;
```

In this code the contents of the two catalogs "test_code" and "production code" are combined into one logical catalog name "program_code". This in turn associated with the filename "code". When the "%include code(run)" is executed, SAS looks to first to the test_code catalog and then to the production_code catalog for the correct version of the element "run.source" to use.

SAS has specific rules for concatenation of catalogs. The concatenation catalog is searched in the order listed on the statement and the first occurrence of the entry found is used. When a catalog is opened for output, it will be created in the first catalog listed. When an element is deleted or renamed, only the first occurrence of the entry is affected. Any time a list of catalog entries is specified, only the first occurrence is given.

The effect of these rules are that when new test code is placed in a catalog ahead of the production catalog. The new test code will be used and the production code left unaffected.

CONCLUSION

Catalogs are excellent tools to use in the management, development, and testing of traditional data step code. All code (source statements, formats, macros, and parameters) are stored in a single file – a SAS catalog. Moving the catalog file moves everything needed to execute the program. The key statement is the use of catalogs is the file name statement with the catalog option.

REFERENCES

SAS Institute Inc., SAS LANGUAGE: CONCEPTS, Cary, NC, SAS Institute Inc., 1999, 554 pages.

SAS Institute Inc., SAS LANGUAGE: REFERENCE, Cary, NC, SAS Institute Inc., 1999, 554 pages.

ACKNOWLEDGMENTS

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.

DISCLAIMER

This paper report the results of research and analysis undertaken by Census Bureau staff. It has undergone a more limited review by the Census Bureau than its publications. This report is released to inform interested parties and to encourage discussion.

CONTACT INFORMATION

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

David D. Chapman
Frame Development Staff
Economic Statistical Methods and Programming Division
US Census Bureau
Washington, DC 20233-6500
Work Phone: (301) 763-4904
Email: david.d.chapman@census.gov