**EXECUTING SAS UTILITIES FROM A MENU-DRIVEN SYSTEM**

Janet C. Lind, San Diego Data Processing Corporation

**INTRODUCTION:**

How many times has the Information Center User or the casual SAS user wanted to perform a simple utility function but could not remember the PROC or the syntax required by the PROC? The user either asks his co-worker if they have done the function or goes to the SAS BASICS manual and tries to find the correct section and then reads through all the parameters. San Diego Data Processing Corporation found that there were several utility functions for which all SAS Users had a need and that much time and many key strokes could be saved by designing some general purpose menus and panels. The original menus and panels were designed for SAS/GRAPH. However those users also required non-graphic utility functions such as PROC SORT, DATASETS, PRINT and DATA steps. Many of these users did not have the knowledge or desire to learn SAS coding. The SAS support personnel did not have the time to sit down with every user and write the SAS code and TSO commands necessary to perform these functions. Therefore the utility menus and the panels evolved from a need to support SAS/GRAPH and Data Entry with simple functions such as displaying directories, observations and sorting to more complex functions such as changing variable lengths, changing from numeric to character, adding formats.

**SAS PRIMARY OPTION MENU:**

The SAS Menu is first presented to the user. For purposes of this paper, the SASTUTIL option will be discussed in detail. However, the other options will be described briefly to illustrate why a utility menu was necessary. Most of these options on this menu require little or no SAS or TSO background.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DATA</td>
<td>SAS DATA definition and Data Entry entry</td>
</tr>
<tr>
<td>2 UTIL</td>
<td>User-defined utility function</td>
</tr>
<tr>
<td>3 PRINT</td>
<td>SAS PRINT, SAS REPORT GENERATION</td>
</tr>
<tr>
<td>4 DATA</td>
<td>SAS DATA definition and Data Entry Entry</td>
</tr>
<tr>
<td>5 DATA</td>
<td>SAS DATA definition and Data Entry Entry</td>
</tr>
</tbody>
</table>

**SAS UTILITY MENU:** Option U

The user must first decide the type of function he wishes to perform and then determine if the function applies to a SAS variable, a SAS Data Set, or a SAS Data Library. All the SAS panels carry the name of the current (last referenced) SAS Data Library and SAS Data Set so they will not have to be rekeyed. All processing is performed in foreground and LOG/PRINT output is directed to the terminal. All functions have SPP HELP panels to display hints for using the selected utility.
DISPLAY FUNCTIONS:

DD - Display the directory of the SAS data sets in a SAS Data Library. A hardcopy printer list can be requested.

DO - Display the variable names in a SAS data set and all the observations. A hardcopy printer list can be requested.

SAS DATA SET:

CC - Copy a SAS Data Set to another SAS Data Set and executing user written SAS code.

CD - Copy multiple SAS Data Sets to the same or another SAS Data Library.

CS - Copy a SAS Data Set to another SAS Data Set. Observations may be selected or deleted based on variables containing specific data.

DS - Delete a SAS Data Set(s) from a SAS Data Library.

RS - Rename or exchange SAS Data Set names.

SS - Sort a SAS Data Set.

SAS VARIABLE NAMES:

AV - Add a new variable name(s) to an existing SAS data set.

DV - Delete SAS variable name(s) and data from a SAS Data Set.

FV - Add/change print "FORMATs", read "INFORMATs" and/or labels.

MV - Modify variable length, or type.

RV - Rename a SAS variable name and/or change label.

SAS DATA LIBRARY:

CL - Copy a SAS Data Library to another SAS Data Library.

CR - Create or reorganize a SAS Data Library.

DISPLAY FUNCTIONS:

The DISPLAY functions are listed first because they are the most heavily used and may be necessary before using any of the other Utility functions. They are executed immediately without the display of another panel.

The "DD" displays the directory. The SAS code generated is:

PROC CONTENTS DATA=ddname. ALL NO SOURCE NODS;

The "DO" displays a variable name list and observation list. The SAS Code generated is:

PROC CONTENTS DATA=ddname.sasdata NO SOURCE;
PROC PRINT DATA=ddname.sasdata;

CC - COPY SAS DATA SET EXECUTING USER SAS CODE

This panel provides for copying a SAS Data Set executing user SAS code. The SAS code can either be on the panel or in an OS data set (sequential or PDS). This is used for simple DATA step statements such as IF, SET, DELETE, OUTPUT. The DATA and SET statements are automatically generated by the dialog. The user SAS code is then appended after the SET statement.

DATA ddname.sasdata;
SET ddname.sasdata;
User SAS code;

PROC DATA LIBRARY NAME DDNAME;
**CD - COPY MULTIPLE SAS DATA SETS**

This panel provides for copying multiple SAS data sets to the same or a different library. There is also an option to copy a spreadsheet screen data which requires a unique parameter on the DATA statement (TYPE=FS Scandinavian). The SAS code generated is:

```sas
DATA inddn.indatal (type);
SET outddn.outdatal;
```

**CS - COPY SAS DATA SET; SELECT, DELETE OBSERVATIONS**

This panel provides for copying a SAS Data Set with a simple selection criteria on the panel. The selection criteria is especially useful for deleting observations with missing values. The SAS code generated is:

```sas
DATA ddname.sasdata;
SET ddname.sasdata;
Optional selection criteria;
```

**DS - DELETE SAS DATA SETS**

This panel provides for deleting SAS Data Sets and releasing unused space. The SAS code generated:

```sas
PROC DATASETS DDNAME=ddname;
DELETE sasdatal sasdatan;
PROC RELEASE DDNAME=ddname;
```

**RS - RENAME/EXCHANGE SAS DATA SETS**

This panel provides for renaming or exchanging SAS Data Set names. The SAS code generated:

```sas
PROC DATASETS DDNAME=ddname NOLIST;
CHANGE/EXCHANGE sasoldl ~ sasnewl sasoldn = sasnewn;
```

---

555
### SS - SORT A SAS DATA SET

This panel provides for sorting SAS Data Sets. The input and output name can be the same. SAS code generated:

```sas
PROC SORT DATA=ddname.sasdata;
  OUT=ddname.sortdata;
  BY (DESCENDING) variable1 (DESCENDING) variables;
```

### DV - DELETE SAS VARIABLES

This panel provides for deleting SAS variables by copying the SAS data set with a DROP statement and copying it back to the original data set. The SAS code generated is:

```sas
DATA WORK.TEMPCOPY (DROP=var1 ... varn);
SET ddname.sasdata;
DATA ddname.sasdata;
SET WORK.TEMPCOPY;
```

### AV - ADD SAS VARIABLES TO A SAS DATASET

This panel adds new SAS variables to an existing SAS data set. The SAS generated code creates default numeric formats and lengths depending on the maximum length of the field.

The SAS code generated is:

```sas
DATA ddname.sasdata;
SET ddname.sasdata;
LENGTH variable1 $4 variable2 4 ... ;
FORMAT variable1 format1 variable2 format2 ... ;
LABEL variable1='labell' variable2='label2' ... ;
```

### DV - FORMAT/INFORMAT/LABEL SAS VARIABLES

This panel provides for adding or changing FORMATS/INFORMATS/LABELS of existing SAS variables. The generated SAS code is:

```sas
PROC DATASETS DDNAME=ddname NOLIST;
MODIFY sasdata;
FORMAT variable1 format1 variable2 format2 ... ;
INFORMAT variable1 informat1 variable2 informat2 ... ;
LABEL variable1='labell' variable2='label2' ... ;
```

### FV - CHANGE EXISTING FORMAT/INFORMAT/LABEL

This panel allows you to change existing formats and labels for SAS variables. You can override the default format types and length. The SAS code generated is:

```sas
PROC DATASET REVENUE RESEX INDEC; INFORMAT
```

### Notes:

- The generated code includes default format lengths and names.
- The `BY` statement sorts variables in descending order by default.
- New variables are added with default formats and lengths.
- Existing variable formats can be changed or added.
- Labels can be specified for variables.
- The `PROC SORT` statement can be modified to suit specific sorting needs.
- The `PROC DATASETS` statement is used for adding, changing, or deleting variables.
- The `LENGTH` statement specifies lengths for new variables.
- The `FORMAT` statement assigns default formats to variables.
- The `INFORMAT` statement assigns default INFORMATs to variables.
- The `LABEL` statement assigns labels to variables.

---

556
MV - MODIFY LENGTH/TYPe OF SAS VARIABLES

This panel provides for changing SAS variable lengths, changing type from character to numeric or numeric to character, change requested. The generated SAS code is not listed because of the length. The general method is:

Use a DATA step create temporary variables from the old variables, assigning the new lengths and formats, dropping old variable names. If there is no numeric length change use PROC DATASETS to rename the variables back to the original names. If there is a numeric length change, use a DATA step to copy the temporary variables back to the original variable names, dropping the temporary variable names and reassigning lengths and formats.

PROC MODIFY DATASETS DDNAME=ddname NOLIST;
MODIFY sasdata;
LABEL oldvar1 = 'label1' ...
RENAME oldvar1 = newvar1 ...

PROC DATASETS DDNAME=ddname NOLIST;
MODIFY sasdata;
LABEL oldvar1 = 'label1' ...
RENAME oldvar1 = newvar1 ...

CL - COPY SAS DATA LIBRARY

This panel provides for copying SAS data libraries with the option of copying all, selecting or excluding SAS data sets. The SAS code generated is:

PROC COPY IN=iddname OUT=oddname;
SELECT/EXCLUDE sasdata...

CR - CREATE/REORGANIZE SAS DATA LIBRARY

This panel provides allocation of a SAS Data Library. A CLIST allocates space.

PROC LIBRARY CREATE/REORGANIZE;
LIBRARY sasdata;
NUMBER OF TRACKS ...
PREFIX, USERID, SECRET...

RV - RENAME/LABEL SAS VARIABLES

This panel provides for renaming and relabeling SAS variables. The SAS code generated:

PROC DATASETS DDNAME=ddname NOLIST;
MODIFY sasdata;
LABEL oldvar1 = 'label1' ...
RENAME oldvar1 = newvar1 ...

PROC DATASETS DDNAME=ddname NOLIST;
MODIFY sasdata;
LABEL oldvar1 = 'label1' ...
RENAME oldvar1 = newvar1 ...

PROC LIBRARY CREATE/REORGANIZE;
LIBRARY sasdata;
NUMBER OF TRACKS ...
PREFIX, USERID, SECRET...

If reorganization is chosen, the SAS data library is copied to a temporary library and a second panel is then displayed showing actual track usage.
IMPLEMENTATION:

This SAS Utility Panel System is implemented using the IBM ISPF Dialog Management System. The Dialog Manager drives SAS by means of a high level CLIST which allocates all the SAS system files and displays the SAS MENU panel. Lower level CLISTs and panels are used for inputting and editing of variable values and file names. These variable values are used as input to the File Tailoring portion of the Dialog Manager which builds SAS code depending on the variable values entered. The generated SAS code file is allocated to FILE SYSIN. SAS files PTF1IF001(LOG) and PTF12F001(PRINT) are allocated to the terminal. The SAS OPTIONS NONOTES SYSIN=SYSIN are passed to SAS at execution time. A SAS (GO form) CLIST is then invoked to enter SAS.

An example of a dialog to delete SAS Variable Names and data follows:

1. The SASDELEV CLIST is invoked from the SAS Utility Menu when option "DV" is selected.
2. The SAS Delete Variable panel is displayed and all information is gathered from the screen.
3. The SAS code is generated using File Tailoring skeleton language.
4. SAS is then invoked using the nested CLIST SASGO.

CONCLUSION:

This method of implementation was chosen because the SAS users and implementor were already familiar with ISPF Dialogs. The major drawback is the time it takes to enter and exit SAS. In order to keep this time at a minimum, all SAS allocations are done at one time when processing the top SAS Menu. Time is also saved in that the user does not make SAS syntax errors and the user does not need to know or use TSO commands.

For installations that do not have the ISPF Dialog Manager or the technical personnel available to write ISPF dialogs, a similar application could be accomplished in SAS using PROC FSEDIT with screen modification to collect the information in temporary SAS Data sets. A SAS program could then generate a PUT file containing SAS code to perform the requested function.

Regardless of implementation, all SAS users whether novice, casual or experienced can benefit from a menu-driven interactive SAS utility application.

For further information contact:

Janet C. Lind
San Diego Data Processing Corp.
5975 Santa Fe Street
San Diego, Ca. 92109
(619) 236-5977