Paper 040-29

Helpful Undocumented Features in SAS®

Wei Cheng, ISIS Pharmaceuticals, Inc., Carlsbad, CA

ABSTRACT

The SAS OnlineDoc® and the built-in SAS® System Help contains comprehensive documentation and references to help SAS users. But we can still find some features that are not well documented or are not documented at all. This paper will try to document some of these undocumented features, so you can use them in the SAS programming.

INTRODUCTION

When we read other people's programs, we see options that we can't find documentated in the SAS OnlineDoc. When we read some papers, most often written by the developers from the SAS Institute, undocumented features are unveiled. When we ask questions or report problems to the SAS Institute technical support, we receive solutions from them that are not documented.

This paper will document some of these undocumented features. Some of them will help SAS programmers to improve their programming efficiency, some will help SAS programmers understand the SAS system better, and some will remove the surprise when SAS programmers see the unexpected behaviour of the SAS system.

First, let's generate some dummy data for testing purpose:

```
data testdata1;
  do id = 1 to 100;
    x = ceil (ranuni(0) * 100);
    charx = put (x, z2. -L);
    output;
  end;
run;

data testdata2;
  do id = 1 to 100;
    y = ceil (ranuni(0) * 100);
    output;
  end;
run;
```

FUNCTION MONOTONIC( ) IN PROC SQL

The automatic variable _N_ in DATA step processing counts the number of times the DATA step begins to iterate. It's very useful when you need the iteration number from the DATA step. Since PROC SQL uses a relational database concept that is different from the DATA step, we can't get the iteration number from the PROC SQL procedure. An undocumented function, MONOTONIC( ), in PROC SQL that can generate very similar result as the _N_ in DATA step. Look at the following example:

Example 1:

```
proc sql;
  select monotonic() as rowno, *
  from testdata2
  where monotonic() le 10;
quit;
```

The above program will generate the output:

rowno	id	y
1	1	66
2	2	32
3	3	10
4	4	24
5	5	50
6	6	73
7	7	40
8	8	45
9	9	88
10	10	65

So we can treat the MONOTONIC( ) function in PROC SQL as the _N_ in DATA step if we need to use the row number of the table in PROC SQL.

COLON (:) MODIFIER COUNTERPART IN PROC SQL

In the DATA step, a colon (:) modifier after any operator can be used to compare only a specified prefix of a character string. In PROC SQL, the following truncated string comparison operators are available: EQT, GTT, LTT, GET, LET, NET, etc.

These operators compare two strings after making the strings the same length by truncating the longer string to the same length as the shorter string.

Example 2:

```
proc sql;
  select *
  from testdata1
  where charx eqt '0';
quit;
```

The above program will generate the output:

```
      id         x  charx
--------------------------
       4         2    02
      36         3    03
      39         6    06
      46         9    09
      48         3    03
      55         9    09
      57         5    05
      75         4    04
      84         2    02
      90         9    09
```

These operators will be documented in the next release of SAS.

INDEXW( ) BEHAVIOUR

The SAS OnlineDoc does not document what happens when the second argument for INDEXW( ) is blank. A little birdie from the SAS Institute unveiled the mystery from the source code. The rule is:

If the second argument contains only blanks or has a length of zero, then:
If the first argument contains only blanks or has a length of zero, INDEXW returns 1;
Otherwise, INDEXW returns 0.

Example 3:

```
data _null_;
  result1 = indexw('         ',' '); put result1 =;
  result2 = indexw('         ', ''); put result2 =;
  result3 = indexw('','          '); put result3 =;
  result4 = indexw('Any Chars',' '); put result4 =;
run;
```

The above program will generate the log:

```
result1=1
result2=1
result3=1
result4=0
```

FILE NAME LIMIT IN PROC EXPORT

When using PROC EXPORT to export SAS data sets to Excel files, the length of the whole name of the Excel file including the path cannot exceed **64** characters. Usually you won't have an Excel file name that exceeds 64 characters, but you may have a long path for the location of the file. You either need to change the location of the file or use the driver mapping to make the path shorter.

"SECRET" SYSTEM OPTIONS

If you want to view the "secret" SAS system options, add the undocumented option INTERNAL to the PROC OPTIONS statement:

Example 4:

```
proc options internal; run;
```

The above program will display the options below in the SAS LOG file with descriptions:

```
 Internal Portable Options:

  NOAUTOSP           Do not load stored DATA step programs
  NOBOOTSTRAP        SASHELP.CORE is available at SAS Session Startup
  BUFMAX=0           Maximum page size for SAS data sets
  BUFOBS=80          Number of OBS in page of SAS data set
  CBUFSIZE=0         Size of buffers to use for SAS catalogs
  CGOPTIMIZE=3       >>Option Description Needed<<
  COLUMNS=80         >>Option Description Needed<<
  DEBUG=JUNK         Internal debugging specification
  DSOPTIONS=         Internal DATA step execution options
  NOFORMATLOG        Do not format the log file
  ……
```

DISPLAY THE OPTIONS BY GROUP

If you want to display the options by group, specify a group-name to the option GROUP in the PROC OPTIONS statement:

Example 5:

```
proc options group = inputcontrol; run ;
```

The above program will display the options below in the SAS LOG file with descriptions:

```
  NOCAPS                  Do not translate source input to uppercase
  NOCARDIMAGE             Do not process SAS source and data lines as 80-byte records
  INVALIDDATA=.           Missing value to assign when invalid numeric data encountered on
                          input
  S=0                     Length of source statements and data lines
  S2=0                    Length of secondary source statements, such as input from a
                          %INCLUDE statement, an AUTOEXEC file, or an autocall macro file
  SEQ=8                   Number of digits in numeric portion of the sequence field
  NOSPOOL                 Do not write SAS statements to a utility data set in the WORK data
                          library
  YEARCUTOFF=1920         Cutoff year for DATE and DATETIME informats and functions
  NUMKEYS=12              Number of function keys. (default on machine keyboard)
  NUMMOUSEKEYS=3          Number of keys (buttons) on mouse
  PFKEY=(WIN)             Key definitions to map
  NOWEBUI                 Do not use web user interface enhancements for supporting hover
                          mode and single-click expand/activation.
```

Acceptable group-names under SAS V8.2 are:

COMMUNICATIONS
ENVDISPLAY
ENVFILES
ERRORHANDLING
EXECMODES
EXTFILES
GRAPHICS
INPUTCONTROL
INSTALL
LANGUAGECONTROL
LISTCONTROL
LOGCONTROL
LOG_LISTCONTROL
MACRO
MEMORY
ODSPRINT
PERFORMANCE
SASFILES
SORT

DISPLAY THE ENGINES

If you want to view the engines available, there is an undocumented procedure.

Example 6:

```
proc nickname; run;
```

The above program will display a list of engines in the SAS LOG file with descriptions:

```
        Nickname  Module    Type   Fileformat   Description

     M  ACCESS99  SASECRSP  ENG                 Read engine for CRSP ACCESS97 database
     M  BASE      SASE7     ENG    7            Base SAS I/O engine
P    M  BMDP      SASBMDPE  ENG    607          BMDP Save File engine
P    M  CRSPACC   SASECRSP  ENG                 Read engine for CRSP ACCESS97 database
P    M  DB2       SASIODBU  ENG    7            SAS/ACCESS Interface to DB2
P       DBIPRO    SASIOPRO  ENG    7            DBI Prototype engine
P    M  FAMECHLI  SASEFAME  ENG                 Seamless libname interface to FAME db
P    M  ODBC      SASIOODB  ENG    7            SAS/ACCESS Interface to ODBC
P    M  OLEDB     SASIOOLE  ENG    7            SAS/ACCESS Interface to OLE DB
P    M  ORACLE    SASIOORA  ENG    7            SAS/ACCESS Interface to Oracle
……
```

REGISTER THE LOCATION OF THE SAS SYSTEM

If your SAS system is not registered properly during an installation, or if you have installed SAS 8 and SAS 9 on the same machine, you could use an undocumented system option –REGSERVER to register the default location of the sas.exe program. This configuration option can be used in a command prompt:

Example 7:

```
!SASROOT\sas.exe -regserver
```

where !SASROOT is the location of the SAS directory (i.e. C:\Program Files\SAS Institute\SAS\V8).

PREVENTING THE APPEARANCE OF THE POP-UP WINDOWS

An undocumented option **-nosleepwindow** exists that prevents the appearance of the pop-up windows that normally appear when you use the SLEEP and WAKEUP functions. This invocation option can be used in the startup command of your SAS session shortcut, with your RUN command, or in your sasv8.cfg file.

This option will be documented in the next release of SAS.

JOIN METHOD CHOSEN BY PROC SQL

If you want to improve the join performance of programs that use PROC SQL to join tables, you need to know how the PROC SQL query optimiser chooses the join methods. There is an undocumented option _METHOD on the PROC SQL statement that will display the hierarchy of processing methods that will be chosen by PROC SQL. You need to set the SAS System option MSGLEVEL = I to see the internal form of the query plan in the SAS LOG. An undocumented option _TREE will show the hierarchy tree as planned in the SAS LOG.

The PROC SQL execution methods include:

sqxcrta	Create table as Select
sqxslct	Select
sqxjsl	Step Loop Join (Cartesian)
sqxjm	Merge Join
sqxjndx	Index Join
sqxjhsh	Hash Join
sqxsort	Sort
sqxsrc	Source Rows from table
sqxfil	Filter Rows
sqxsumg	Summary Statistics (with GROUP BY)
sqxsumn	Summary Statistics (not grouped)
sqxuniq	Distinct rows only

Example 8:

```
options msglevel = I;

proc sql _method
         _tree;
  select t1.id
        ,x
        ,y
        ,z
  from testdata1 as t1
      ,testdata2 as t2
      ,junk._totest as t3
  where t1.id = t2.id and
        t2.id = t3.id;
quit;
```

The above program will display the query plan below in the SAS LOG file:

```
sqxslct
 sqxjhsh
  sqxjhsh
   sqxsrc( ORK.TESTDATA2(alias = T2) )
    sqxsrc( JUNK._TOTEST(alias = T3) )
     sqxsrc( WORK.TESTDATA1(alias = T1) )

Tree as planned.
                                  /-SYM-V-(t1.id:1 flag=0001)
                        /-OBJ----|
                       |         |--SYM-V-(t1.x:2 flag=0001)
                       |         |--SYM-V-(t2.y:2 flag=0001)
                       |          \-SYM-V-(t3.z:2 flag=0001)
             /-JOIN---|
            |          |                              /-SYM-V-(t2.id:1 flag=0001)
            |          |                    /-OBJ----|
            |          |                   |         |--SYM-V-(t2.y:2 flag=0001)
            |          |                   |         |--SYM-V-(t3.id:1 flag=0001)
            |          |                   |          \-SYM-V-(t3.z:2 flag=0001)
            |          |          /-JOIN---|
            |          |         |         |                              /-SYM-V-(t2.id:1
flag=0001)
            |          |         |         |                    /-OBJ----|
            |          |         |         |                   |          \-SYM-V-(t2.y:2
flag=0001)
            |          |         |         |          /-SRC----|
            |          |         |         |         |          \-TABL[WORK].testdata2 opt=''
            |          |         |         |--FROM---|
            |          |         |         |         |                    /-SYM-V-(t3.id:1
flag=0001)
            |          |         |         |         |          /-OBJ----|
            |          |         |         |         |         |          \-SYM-V-(t3.z:2
flag=0001)
            |          |         |         |          \-SRC----|
            |          |         |         |                    \-TABL[JUNK]._totest opt=''
            |          |         |         |--empty-
            |          |         |         |          /-SYM-V-(t2.id:1)
            |          |         |          \-CEQ----|
            |          |         |                    \-SYM-V-(t3.id:1)
            |          |--FROM---|
            |          |         |                    /-SYM-V-(t1.id:1 flag=0001)
            |          |         |          /-OBJ----|
            |          |         |         |          \-SYM-V-(t1.x:2 flag=0001)
            |          |          \-SRC----|
            |          |                    \-TABL[WORK].testdata1 opt=''
            |          |--empty-
            |          |          /-SYM-V-(t2.id:1)
            |           \-CEQ----|
            |                    \-SYM-V-(t1.id:1)
 --SSEL---|
```

The SELECT module (sqxslct) gets its input records from the Hash Join Module (sqxjhsh), which gets its input from three sources (sqxsrc).

DATA SETS DISAPPEAR

When you create data sets that start with _to, like _total, _top, they will not be visible to the SAS EXPLORE window, but, they are in the directory. You can treat them the same as other data sets, even though you can't see them in the SAS EXPLORE window. This is because when the SAS system operates, _to is used as the prefix to the temporary data sets and the temporary data sets are intended to be hidden from the SAS users. So don't be surprised when you can't see the data sets that you just generated in your program.

Example 9:

```
libname junk "C:\junk";

data junk._totest;
  do id = 1 to 100;
    z = ceil (ranuni(0) * 100);
    output;
  end;
run;
```

CHANGE THE SYMBOL ON THE KAPLAN-MEIER PLOT

PROC LIFETEST can generate a high-resolution Kaplan-Meier Curve. The CENSOREDSYMBOL option specifies the symbol value for the censored observations.

When you specify the strata levels in the STRATA statement, you always have the same symbol on the curves for censored observations for all the strata levels. But sometimes we prefer having different symbols for different strata levels. This can be done by an undocumented option for the CENSOREDSYMBOL option. When you put CENSORSYMBOL = ' ' in the PROC LIFETEST statement, you can change the symbols used for censored observations in the plots by SAS/GRAPH SYMBOL statement.

Example 10:

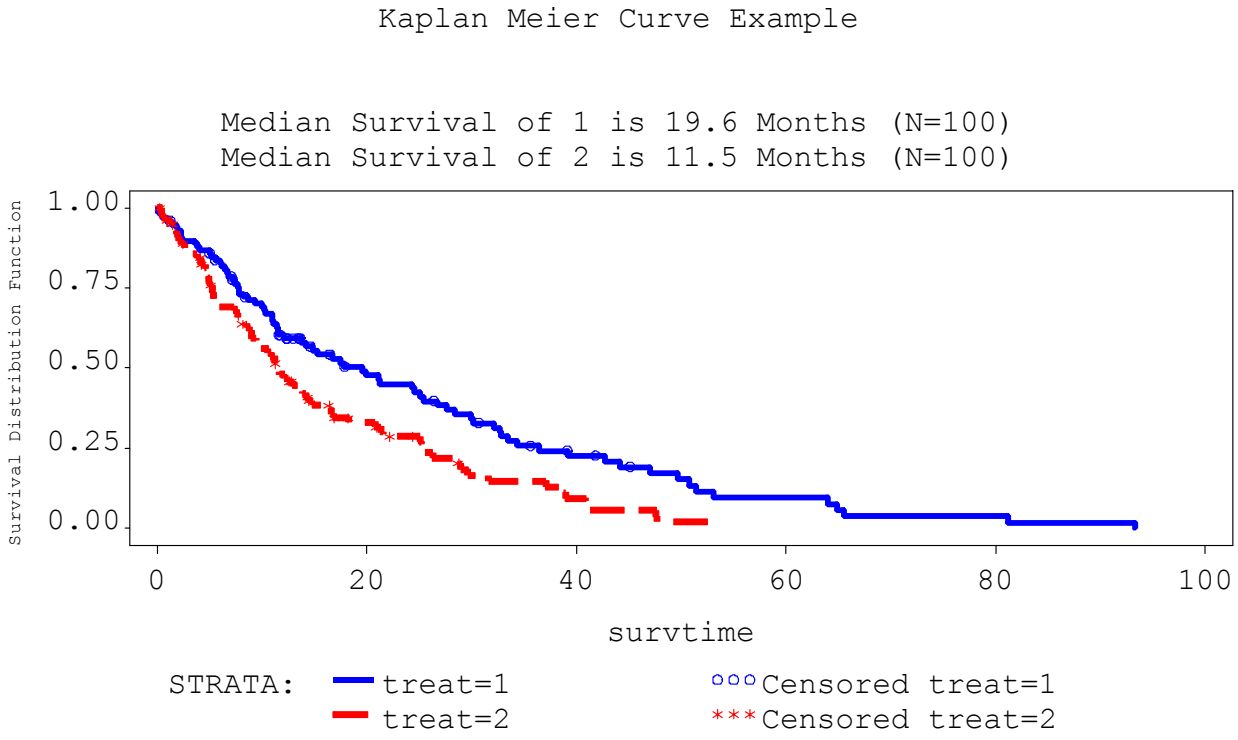
```
data survsamp;
 retain seed 20020824 n 1 p 0.2;
  do treat = 1 to 2;
   do patid = 1 to 100;
    call ranbin(seed,n,p,censor);
    call ranexp(seed,xi);
    if treat = 1 then
     survtime = xi / (log(2) / 15);
    else
     survtime = xi / (log(2) / 10);
    output;
   end;
  end;
 drop seed n p xi;
 format patid z3.;
run;

symbol1 line  = 1
  color = blue
  width = 1
  v     = circle;

symbol2 line  = 4
  color = red
  width = 1
  v     = triangle;

proc lifetest data = survsamp    noprint
     plot = (s) censorsymbol = ' ';
  time survtime * censor(1);
  strata treat;
run;
```

The above program will generate the Kaplan-Meier curve:




```
P value from Log-Rank Test: 0.0023
SOURCE: EXAMPLE PROGRAM.SAS WCHENG SASv8.2 (05JAN2004 19:29)
```

CONCLUSION

Since these features are undocumented, we should use them with caution. It's better to test the program thoroughly or consult with the SAS Institute technical support before we apply them to our programming. They may remain undocumented because they are not tested thoroughly, or they perform badly in some combination with other options, or on some host systems, etc. The programs in this paper are tested on SAS version 8.2 for Windows. Most of the undocumented features covered in this paper can be adapted to other operating systems.

REFERENCES

SAS Institute, Inc., SAS OnlineDoc, Version 8, Cary, NC: SAS Institute, Inc., 1999.

SAS Institute, Inc., SAS System Help, Version 8, Cary, NC: SAS Institute, Inc., 1999-2001.

SAS Institute, Inc., SAS Notes SN-000258, Cary, NC: SAS Institute, Inc., 2001

SAS Institute, Inc., SAS Notes SN-001474, Cary, NC: SAS Institute, Inc., 2001.

SAS Institute, Inc., SAS Notes SN-001714, Cary, NC: SAS Institute, Inc., 2001.

SAS Institute, Inc., SAS Notes SN-007264, Cary, NC: SAS Institute, Inc., 2002.

SAS Institute, Inc., SAS Notes SN-008234, Cary, NC: SAS Institute, Inc., 2003.

SAS Institute, Inc., SAS Notes SN-008900, Cary, NC: SAS Institute, Inc., 2002.

SAS Institute, Inc., SAS Notes SN-010363, Cary, NC: SAS Institute, Inc., 2003.

SAS Institute, Inc., SAS Notes SN-011065, Cary, NC: SAS Institute, Inc., 2003.

Kent, P. (1995), "SQL Joins - The Long and The Short of It" Proceedings of the Twentieth Annual SAS Users Group International Conference, 20, 206-215.

Holland, P. “Formats, Options, and Functions” Views Issue 21, 1st Quarter 2003.

ACKNOWLEDGMENTS

I would like to thank Paul Kent and Julia Schelly of SAS Institute for their review and comments. I would also like to acknowledge the contributors on SAS-L.

CONTACT INFORMATION

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

Wei Cheng
Isis Pharmaceuticals, Inc.
2292 Faraday Ave.
Carlsbad, CA 92008
Work Phone: (760) 603-3807
Fax: (760) 603-2588
Email: wcheng@isisph.com
Web: http://www.prochelp.com