

## Some Questions and Solutions From a SAS® Help Desk

Stephen B Taubman, Federal Reserve Board, Washington, DC

### ABSTRACT

I have been answering questions at an in-house SAS Help desk for several years. Whenever a solution involved writing SAS code, I saved the code for possible future use. Over the years I have accumulated many SAS programs. I will now present some interesting questions along with the solutions. Some of the solutions were obtained with help from SAS Institute Technical Support or the comp.soft-sys.sas news group. The vast majority of questions which I received were related to economic or financial SAS applications and executed on a UNIX SOLARIS SUN operating system for Release 6.11. However, most of the information in this paper can be used in more general applications on any operating system. A SAS novice should understand some of this paper.

### EXAMPLE 1

```
/*Example 1 convert numeric missing values to -99999 */
data one;
input chr $ a b c;
cards;
a 2 . 9
b . . 22
. . 3 .
d 8 . .
;
data two;
set one;
drop i;
array x(*) _numeric_;
do i= 1 to dim(x);
if x(i) = . then x(i)=-99999;
end;
proc print data=two;
title 'missing values converted to -99999';
```

### Example 1 output

missing values converted to -99999				
OBS	CHR	A	B	C
1	a	2	-99999	9
2	b	-99999	-99999	22
3		-99999	3	-99999
4	d	8	-99999	-99999

### EXAMPLE 2

```
/* Example 2 form cumulative product of input */
data one;
input x;cards;
3
6
.5
100
;
data t;set one;
retain s;
if _n_ eq 1 then s=x;
```

```
else s=s*x;
proc print;
title 'cumulative product';
```

### Example 2 output

cumulative product		
OBS	X	S
1	3.0	3
2	6.0	18
3	0.5	9
4	100.0	900

### EXAMPLE 3

Remove SAS datasets no longer active from the work space. SAS supplies a command found in the SASROOT directory to accomplish this.

### Example 3 Solution

From a UNIX prompt, enter :

```
SASROOT_dir /utilities/bin/cleanwork /tmp
where SASROOT_dir is the SAS root directory, and /tmp
is the work directory for this example.
```

### EXAMPLE 4

```
/* Example 4 create a flat file from a SAS dataset */
data one; ;
/* create sample data */
do i = 1 to 6;
x=ranuni(-1);
xx=x*2;
xxx=x*3;
chr='lklklk';
output;
end;run;
options mprint;
%flatfile(lib=work, dsn=one, file=outfill);
/* flatfile macro is available from SAS Institute */
```

### Example 4 output

```
outfill
1 0.0155966 0.03119321 0.04678981 lklklk
2 0.39924458 0.79848916 1.19773374 lklklk
3 0.92313608 1.84627217 2.76940825 lklklk
4 0.00496401 0.00992802 0.01489202 lklklk
5 0.29958711 0.59917422 0.89876133 lklklk
6 0.94001905 1.88003809 2.82005714 lklklk
```

obtain variable order from the following statement in the log:

```
PUT I BEST10. +1 X BEST10. +1 XX BEST10. +1 XXX
BEST10. +1 CHR $ 6. +1 ;
```

### EXAMPLE 5

```
/*Example 5 print out a SAS date as fiscal year using proc
format */
proc format;
value fy
'01Oct86'D-'30Sep87'D='FiscalYear 87'
```

```

'01Oct87'D-'30Sep88'D='FiscalYear 88'
'01Oct88'D-'30Sep89'D='FiscalYear 89'
'01Oct89'D-'30Sep90'D='FiscalYear 90'
'01Oct90'D-'30Sep91'D='FiscalYear 91'
'01Oct91'D-'30Sep92'D='FiscalYear 92'
'01Oct92'D-'30Sep93'D='FiscalYear 93'
'01Oct93'D-'30Sep94'D='FiscalYear 94'
;
data qq;
d1='2Oct88'D;
d11=d1;
d2='21Dec86'D;
d22=d2;run;
title 'date and fiscal date';
proc print;
var d11 d1 d22 d2;
format d1 fy. d2 fy. d11 date7. d22 date7.;

```

**Example 5 output**

date and fiscal date				
OBS	D11	D1	D22	D2
1	02OCT88	FiscalYear 89	21DEC86	FiscalYear 87

**EXAMPLE 6**

```

/*Example 6 put output from a UNIX command into a
SAS dataset */
filename aaa pipe "ps -ef|grep sbt";
/* UNIX command in filename statement executes.
output from that command (written to standard
output) is read into a SAS dataset(xin) */
data xin; length process $ 70; infile aaa pad lrecl=85;
input process $char70.;
proc print;title 'output from pipe example';

```

**Example 6 output**

output from pipe example				
OBS	PROCESS			
1	m1sbt00	4743	1 0 12:04:21 ?	0:00 xpositit
2	m1sbt00	5213	1 0 12:27:45 ?	0:32 /interleaf/ileaf5/sun
3	m1sbt00	5624	4790 1 13:12:23 pts/28	0:0 0/opt/sas/sas611
4	m1sbt00	4736	1 0 12:04:20 ?	0:00 xbiff

**EXAMPLE 7**

```

/* Example 7 submit a noninteractive SAS job with one
macro variable defined in the SAS command statement*/
/* submit with command 'sas -sysparm 4 code1.sas'
where code1.sas follows*/
%put &sysparm ' this is input'; run;
data q;
nputparm=&sysparm;run;
proc print;
title 'output of noninteractive SAS job with one input
parameter';

```

**Example 7 output**

output of noninteractive SAS job with one input parameter

OBS	NPUTPARM
1	4

**EXAMPLE 8**

```

/*Example 8 retrieve UNIX environment variable into a
SAS variable*/
data one;
printnam=sysget('PRINTER');
proc print;
title 'PRINTER environment variable'; run;

```

**Example 8 output**

PRINTER environment variable	
OBS	PRINTNAM
1	aprnt1

**EXAMPLE 9**

```

/* Example 9 form a variable of consecutive monthly
dates */
%macro datevar
(data=listdate,freq='month',num=,month1=1,day1=1,year
1=1900);
* construct sas variable of dates;
data &data;
format datelist date9.;
drop _i;
do _i=1 to &num;
datelist=intnx(&freq,mdy(&month1,&day1,&year1),(_i-1
));
output;
end;
%mend datevar;
%datevar(freq='month',num=3,year1=1976);
proc print data= listdate;
title 'a list of monthly dates';
format datelist date9.;
run;

```

**Example 9 output**

a list of monthly dates	
OBS	DATELIST
1	01JAN1976
2	01FEB1976
3	01MAR1976

**EXAMPLE 10**

```

/*Example 10 convert a flat file to all lower case*/
filename inn 'origcase';
filename outt 'lowcase';
data one;
length v1-v2 $72;
file outt;
infile inn lrecl=72 pad;
input @1 v1 $char72.;
v2=trim(lowcase(v1));
put @1 v2 ;

```

**Example 10 output**

origcase file:  
GET into your voice-mail box (as though you were checking messages).  
Hit 7 to record or change a greeting.  
Hit 11 to record the greeting.  
Hit # to end recording.

lowcase file :  
 get into your voice-mail box (as though you were  
 checking messag  
 hit 7 to record or change a greeting.  
 hit 11 to record the greeting.  
 hit # to end recording.

### EXAMPLE 11

```
/* Example 11 get Catesian product */
data one; input list1 ;
cards;
1
3
9
;
data two; input list2 $ ;
cards;
a
b
d
e
;
proc sql; create table product as select * from
(select distinct list1 as list1 from one),
(select distinct list2 as list2 from two);
quit;
proc print data=product; title 'cartesian product';
```

### Example 11 output

cartesian product		
OBS	LIST1	LIST2
1	1	a
2	1	b
3	1	d
4	1	e
5	3	a
6	3	b
7	3	d
8	3	e
9	9	a
10	9	b
11	9	d
12	9	e

### ACKNOWLEDGMENTS

The author wishes to thank Ray Board and Bruce Gilson for comments throughout the development of this paper. The author can be contacted at :

Federal Reserve Board  
 Mail Stop 096  
 Washington, DC 20551  
 (202)452-2517  
 staubman@frb.gov

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries.® indicates USA registration