

SAS Tips and Tricks

1

TASS
09 FEBRUARY 2018

Contents

2

- **DLCREATEDIR recursive macro**

DLCREATE recursive macro

3

- From December 2017 TASS meeting, we learned:

DLCREATEDIR

```
options DLCREATEDIR; ** ← Turn on the SAS system option  
to create a directory if it does not already exist **;
```

```
libname TASS "C:/TASS";
```

```
data TASS.TEMP;
```

```
    x=1;
```

```
run;
```

```
9  options DLCREATEDIR; ** <-- Turn on the SAS system  
option to create a directory if it  
9  ! does not already exist **;  
10 libname TASS "C:/TASS";  
NOTE: Library TASS was created.  
NOTE: Libref TASS was successfully assigned as follows:  
Engine:          V9  
Physical Name: C:\TASS  
  
12 data TASS.TEMP;  
13     x=1;  
14 run;  
  
NOTE: The data set TASS.TEMP has 1 observations and 1  
variables.  
NOTE: DATA statement used (Total process time):  
real time          1.29 seconds  
cpu time           0.00 seconds  
  
cpu time           0.00 seconds
```

Lisa Eckler CONSULTING INC.

TASS September 2017

DLCREATE recursive macro (2)

4

- We also learned :

Usage:

1. `options NODLCREATEDIR;` **<-- to switch this option off**
2. **DLCREATEDIR** is available from 9.3 on.
3. **Creates a single level only**

```
libname TASS2017 "C:/TASS/2017/September";
```

```
libname TASS2017 "C:/TASS/";
```

```
libname TASS2017 "C:/TASS/2017";
```

```
libname TASS2017 "C:/TASS/2017/September";
```

DLCREATE recursive macro (3)

5

- A possible solution for UNIX:

```
%macro cdir(mylib, mypath);
options dlcreatedir;
  data have_xyzzy;
    orig_var = "&mypath.";
  run;
  proc sql noprint;
    select max(count(orig_var,'/')) into :maxelements from have_xyzzy;
  quit;
  %put number of levels = &maxelements.;

  %do i = 1 %to &maxelements.;
    data _null_;
      length mypat $160.;
      drek = "&mypath.";
      mypat = "";
      do c = 1 to &i.;
        mypat = compress(mypat || '/' || scan(drek,c,'/'));
      end;
      call symput('mypat',mypat) ;
    run;
    %put loop# = &i. path = &mypat.;
    libname &mylib. "&mypat.";
  %end;
%mend;
%cdir(temp3, /sasdata/rsk/prs/devvol02/mytemp1/mytemp2/mytemp3);
```

DLCREATE recursive macro (4)

6

- A possible solution for UNIX:

```
%macro cdir(mylib, mypath);
options dlcreatedir;
  data have_xyzzy;
    orig_var = "&mypath.";

  run;
  proc sql noprint;
    select max(count(orig_var,'/')) into :maxelements from have_xyzzy;
  quit;
  %put number of levels = &maxelements.;

  %do i = 1 %to &maxelements.;
    data _null_;
      length mypat $160.;
      drek = "&mypath.";
      mypat = "";
      do c = 1 to &i.;
        mypat = compress(mypat || '/' || scan(drek,c,'/'));
      end;
      call symput('mypat',mypat) ;
    run;
    %put looper = &i. path = &mypat.;
    libname &mylib. "&mypat.";
  %end;
%mend;
%cdir(temp3, /sasdata/rsk/prs/devvol02/mytemp1/mytemp2/mytemp3);
```

Concatenates recursive string variable

Sets path as a macro variable

Sets local macro variable to count number of levels on the directory

Sets lib reference and creates directory if necessary

And now, on to Debby...

7

Accessing My Stuff in Linux / Unix

Debby Gear
DG Consulting
February 2018

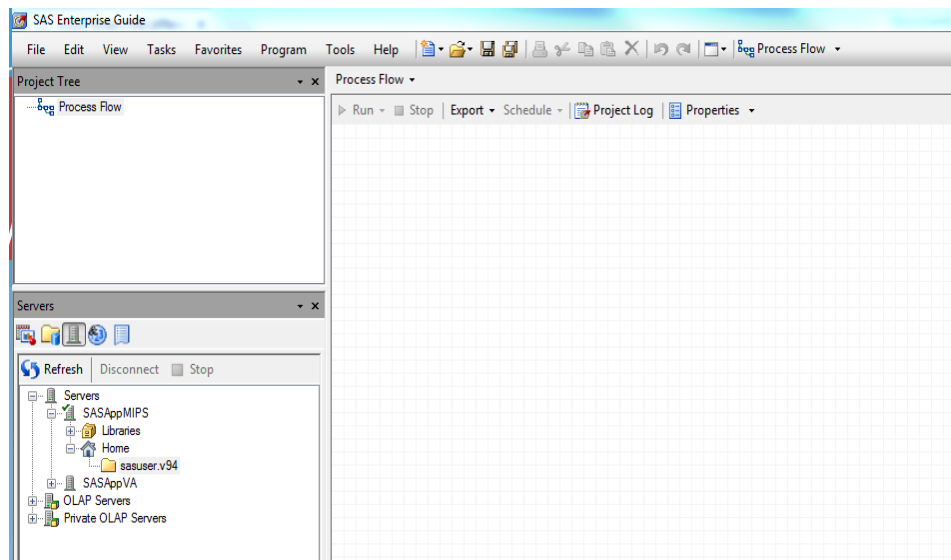
Starting up EG

- ▶ Home directories are usually frowned upon to store anything more than a few KB
- ▶ Typically only startup instructions are placed in this directory
- ▶ Data / code is frequently stored somewhere else, but seems to be inaccessible from EG

Sample of EG Startup,

The data I want is somewhere else, but I can only see my home directory

example location of my Stuff: `/usr/sas/sasdata/GA`



Solution:

Create a soft link to the directory

- ▶ Command from your home directory creates a nickname GA to the location of interest

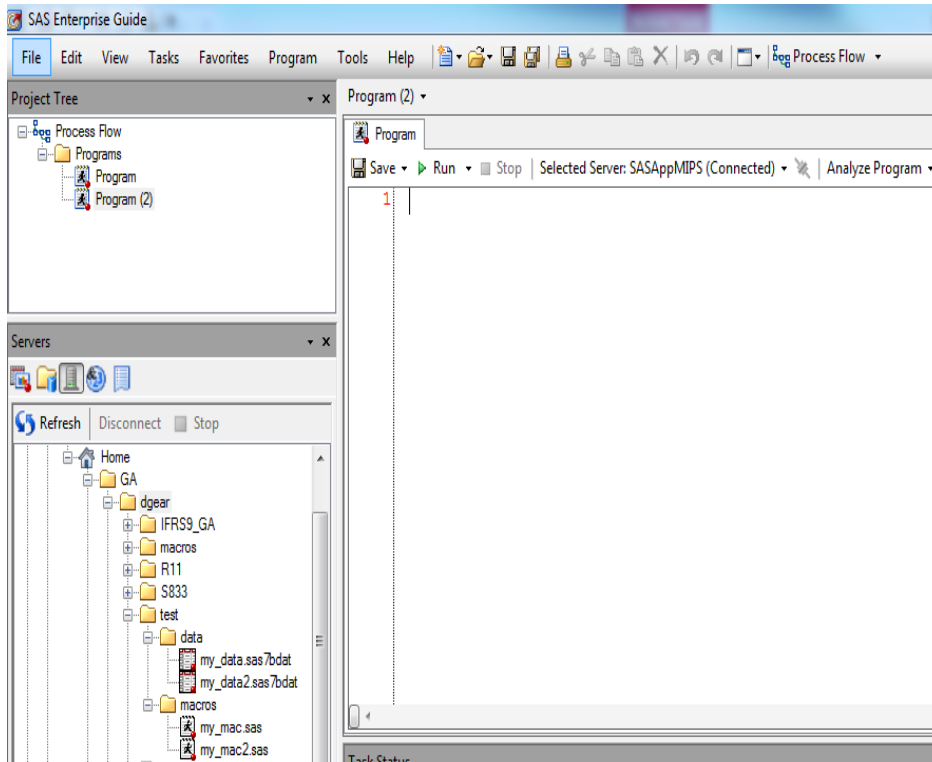
```
ln -s GA /usr/sas/sasdata/GA
```

- ▶ This is a pointer to the directory
- ▶ To see where the link and where it points to

```
ls -la
```

- ▶ GA -> /usr/sas/sasdata/GA

You are now able to access this directory



Contact

Debby Gear
DG Consulting
debby.gear@gmail.com

Will the real John Lam please stand up...

14

KEEP DROP RENAME

Order of Execution?

John Lam
February 9, 2018

```
data test1;  
  a = 6; b=7; c=3; d=1; output;  
  a= 8; b=9; c=4; d=2; output;  
  a= 0; b=10; c=5; d=3; output;  
run;
```



```
data test2;  
  set test1(rename=(d=x) keep=a b  
c x);  
run;
```

```
proc print; run;
```

Obs	a	b	c
1	6	7	3
2	8	9	4
3	0	10	5

```
/* KEEP is executed before RENAME  
*/  
/* variable x does not exist in KEEP  
*/
```

```
data test2;  
  set test1(rename=(d=x) keep=a b  
c d);  
run;
```

Obs	a	b	c	x
1	6	7	3	1
2	8	9	4	2
3	0	10	5	3

```
/* rewrite the previous data step  
with a change: */  
/* variable d is in KEEP */  
/* Result: renamed variable is kept  
*/  
/* Conclusion: KEEP is executed  
before RENAME */
```

```
data test2;  
  set test1(keep=a b c d drop=d);  
run;
```

Obs	a	b	c
1	6	7	3
2	8	9	4
3	0	10	5

```
/* KEEP & DROP the same variable `d`  
*/  
/* Unclear which one executes first in  
this example */  
/* but variable `d` in KEEP is dropped!!!  
- Unexpected result? */
```

```
data test2;  
  set test1(rename=(d=x c=z) keep=a b c d  
drop=z);  
run;
```

Obs	a	b	z	x
1	6	7	3	1
2	8	9	4	2
3	0	10	5	3

```
/* drop a renamed variable ('z') – result: not dropped  
*/
```

```
/* keep a variable ('d') to be renamed – result:  
renamed variable ('x') kept */
```

```
/* Conclusion: RENAME is executed last */
```

Keep, drop & rename, order of execution?

- 1) DROP
- 2) KEEP
- 3) RENAME

Tips:

- 1) Easy way to remember the order: alphabetically (this was a tip from someone in a previous TASS meeting)
- 2) Try not to use KEEP & DROP of the same variable(s) within the same data step