Paper 103-25

Smokin' With UNIX® Pipes

Kimberly J. LeBouton, K.J.L. Computing
Thomas W. Rice, American Honda Motor Company, Inc.

Abstract

The ***FILENAME PIPE*** option can have your SAS® programs "smokin" in no time. ***FILENAME PIPE*** allows you to read in the output of any UNIX operating system command into a SAS dataset. As a SAS dataset, you now have the flexibility of the SAS language, which will also cut down on the amount of UNIX script writing needed.

Managing Data with *ls* Command

Three examples will be provided to demonstrate the use of ***FILENAME PIPE***.

The first example retrieves the output from the UNIX ***ls*** command. The ***ls*** command displays a listing of dataset names. Output 1 is a partial list from the command ***ls -d /appl/icp/**. The ***-d*** option shows only the directories, not the individual files associated with each directory.

Figure 1 shows the ***DATA _NULL_*** step that reads in the dataset names (dsn) and then stores the names in macro variables, named in_1 to in_n. The small macro program, ***MLIBNAME***, uses these macro variables to generate the proper libname statements.

The ***PROC SQL*** creates a table of information about the SAS datasets, similar to the information provided by ***PROC CONTENTS***.

Output 1 Partial Listing from ***ls -d***

```
/appl/icp/icp.crs.sas.data
/appl/icp/icp.hinjosas.data
/appl/icp/icp.pqr2.sas
/appl/icp/icp.sas.camp.accord
/appl/icp/icp.sas.camp.acura
/appl/icp/icp.sas.camp.civic
/appl/icp/icp.sas.camp.cl
/appl/icp/icp.sas.camp.crv
/appl/icp/icp.sas.camp.custcont
/appl/icp/icp.sas.camp.evp
/appl/icp/icp.sas.camp.honda
/appl/icp/icp.sas.camp.integra
/appl/icp/icp.sas.camp.legend
```

Figure 1 Program to read in ***ls -d***

```
filename lsinfo pipe 'ls -d /appl/icp/*';

data _null_;
  infile lsinfo pad end=last;
  input dsn $50.;
  call symput('in'||left(_n_),trim(dsn));
  if last then call symput('ndsn',left(_n_));
run;

%macro mlibname;
  %do i=1 %to &ndsn;
     libname in&i "&&in&i";
  %end;
%mend;

%mlibname run;

proc sql;
  create table icp as
    select *
    from dictionary.tables
    where libname contains 'IN';
quit;
run;
```

```
Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4            32768     20436   38%     1512    10% /
/dev/hd2           811008     76004   91%    24966    13% /usr
/dev/hd9var         57344     30072   48%     2950    21% /var
/dev/hd3            32768     21428   35%      239     3% /tmp
/dev/hd1            57344     45520   21%      827     6% /home
/dev/lv00         2457600    879196   65%    13042     3% /appl
/dev/seilv        4325376   2746124   37%     5069     1% /appl/sei
/dev/lv01          188416     51624   73%     1786     4% /sappl
/dev/worklv      13172736  12754768    4%       29     1% /appl/work
/dev/icplv       34177024   8689560   75%     2367     1% /appl/icp
/appl/icp        34177024   8689560   75%     2367     1% /appl/icp
pa04s:/web         655360    147432   78%        -     -  /amd/pa04s/web
pa04s:/web         655360    147432   78%        -     -  /web
```

Output 2 Listing from ***df -k***

Disk Utilization with *df* Command

To track the amount of disk utilization on two of our disks (/dev/icplv and /dev/worklv), a program was written to retrieve the information from the command ***df -k***. Output 2 displays the information provided by ***df -k***. This data is captured in the ***DATA DF*** step shown in Figure 2. Figure 3 is one observation from the SAS dataset DF.

```
filename df pipe 'df -k';
filename nwork pipe 'ls -d /appl/work/SAS*';
libname worksp '/appl/sei/cb04/';

data _null_;
  infile nwork pad end=last;
  input dsn $30.;
  if last then call symput('nwork',left(_n_));
run;

data df;
  format date mmddyy8. time time5.;
  infile df pad;
  input  @'dev/icplv'
     ibl1024 ifree iperused percent. iiused;
  input  @'dev/worklv'
     wbl1024 wfree wperused percent. wiused;
  njobs=&nwork;
  date=today();
  time=time();
  label ibl1024 ='icp 1024 blocks'
       wbl1024 ='work 1024 blocks'
       ifree   ='icp free'
       wfree   ='work free'
       iperused='icp percent used'
       wperused='work percent used'
       iiused='icp iused'
       wiused='work iused';
run;

proc append base=worksp.master data=df;
run;
```

DATA _NULL_ is another ***FILENAME PIPE***, nwork, that was used to gather information on the number of SAS work directories allocated. This was used to gage the workload on the AIX machine.

PROC APPEND is used to save the data to a permanent SAS dataset. The program is run through a cronjob on a regular basis to refresh the data.

Figure 3 DF SAS Dataset

DATE:	06/06/97
TIME:	9:51
IBL1024:	34177024
IFREE:	8661132
IPERUSED:	0.75
IIUSED:	2334
NJOBS:	15
WBL1024:	13172736
WFREE:	11984672
WPERUSED:	0.1
WIUSED:	296

Figure 2 Program to read ***df -k***

```
Job dahlem.872120760.a will be run at Wed Aug 20 16:46:00 PDT 1997.
```

Output 3 Message from ***at***

Capturing Job # with *at* Command

The ***at*** command allows a user to run a UNIX command at a later time. We used it to submit SAS jobs, which would run in batch at a lower priority. The user could specify now or a later time for the job to run.

Output 3 is the confirmation message that is sent back to the monitor. The job number, which is in bold, can be used to status the job or delete the job if necessary. Once the job is processing this job number goes away, and is replaced by the PID.

Figure 4 captures this message to be used later in a SAS/AF program to be displayed on the message line.

Figure 4 Program to read ***at*** command

```
/* submit batch job */
filename notify pipe
    "nice -50 at &runtime <
       $HOME/batchexec/TC_&titlenb.sas";

/* retrieve system response */
data _null_;
   infile notify pad;
   length info $100;
   input info $100.;
   call symput('mesg',info);
run;

filename notify clear;
```

Summary

The ***FILENAME PIPE*** option allows a SAS programmer access to all operating commands output. This output turned into a SAS dataset, or a SAS macro variable, provides the user an opportunity to perform tasks usually written in UNIX scripts.

References

LeBouton, K. (1997), *The Realities of Downsizing: Moving a SAS® Application from MVS® to UNIX®*. Proceedings of the Twenty Second Annual SAS® Users Group International Conference, Cary, NC: SAS Institute Inc., 658-667.

SAS Institute Inc. (1993), *SAS Companion for UNIX Environment: Language*, Version 6, First Edition, Cary, NC: SAS Institute Inc.

Young, M.L. and Levine, J. (1995), *UNIX® for Dummies®*. Foster City, CA: IDG Books Worldwide, Inc.

About the Authors

Kim LeBouton is an independent consultant with 18 years experience with SAS. Her areas of expertise include base SAS, SAS/STAT, SAS/FSP®, and SAS/AF software. She has a BA degree in Psychology from California State University, Long Beach and an MA degree in Educational Statistics from UCLA. Kim was the Western User of SAS Software 1997 Co-Chair. She is also a SAS Institute Quality Partner.

Kim LeBouton
K.J.L. Computing
3431 Yellowtail Drive
Rossmoor, CA 90720
(562) 594-9235
email: Kim_LeBouton@msn.com

Tom Rice is a Senior Systems Analyst for American Honda Motor Company. He has 10 years of experience in systems development exclusively using the SAS system. He has a BA degree in Mathematics, with an emphasis in Statistics and Computer Science, from Fort Lewis College in Durango, Colorado. He was an invited speaker to SAS Institute's "SAS Software in the Automobile Industry" seminar. Starting in February 2000, Tom Rice will be an employee of SAS Institute.