

Using the FILEVAR= option to read large numbers of data files in one data step

Glenn Millard

Stephen F. Austin State University
Nacogdoches, Texas

Abstract

The FILEVAR option on the INFILE statement is a powerful tool for reading multiple data files, especially if the exact filenames may not be known by the SAS programmer. With this option you can potentially read hundreds of files in one data step.

SAS was used to process snapshots of VMS system statistics without using large amounts of system resources in the process. A set of three standard VMS commands captured the desired information and was executed once each hour. Each command created a separate file each time it was run, resulting in hundreds of files to be read in just a few days. Reading these files into one data set proved to be no problem for SAS. However, careful use of command options is necessary to take full advantage of interrelated features of SAS commands used in the procedure.

The goal of this project was to find an easy way to get snapshots of VMS system usage. Upon making inquiries of VMS users we settled upon using some common VMS commands that showed system usage information. We set up a batch job that executed these commands and then waited one hour before the job was batched again. The batch job consisted of the following commands.

\$! Execute Fortran program that causes system wait for one hour

\$ run home\$acadsup:[millardgd]wait

\$! Get current time and break it into its different parts

```
$ cur_time = F$TIME()
$ day1 = F$EXTRACT(0, 2, cur_time)
$ day = F$EDIT(day1, "TRIM")
$ month = F$EXTRACT(3, 3, cur_time)
$ year = F$EXTRACT(9, 2, cur_time)
$ hour = F$EXTRACT(12, 2, cur_time)
$ minute = F$EXTRACT(15, 2, cur_time)
$ second = F$EXTRACT(18, 2, cur_time)
$ ssecond = F$EXTRACT(21, 2, cur_time)
```

\$! Make the filenames

```
$ fileext = year + month + day + hour + minute + second + ssecond
$ stat1 = "disk$user01:[census.sysstat1]disk" + fileext
$ stat2 = "disk$user01:[census.sysstat1]mem" + fileext
$ stat3 = "disk$user01:[census.sysstat1]user" + fileext
```

\$! Execute the data gathering commands

```
$ show dev du /output='stat1'
$ show memory/output='stat2'
$ show user/output='stat3'
```

Resubmit the batch job to run again

```
$ submit/nolog/restart/queue=bs$batch -      home$acadsup:[millardgd]sysstat.com
$ exit
```

The three VMS show commands write their output to files that have the time as part of the filename to make sure each filename is unique. There is not enough flexibility in these SHOW commands to allow just one file to be written. Therefore the result is that three files are created each hour, making hundreds of files in just a few days.

Our next task was to read these files into a SAS data set so that reports and graphs could be produced. The challenge was whether SAS could potentially read hundreds of files in one data step. The FILEVAR option of the INFILE statement provided the solution to the problem.

Exact filenames to be read are unknown to the SAS program prior to runtime. This is because the filename includes a date as part of the filename. A simple VMS directory command can create a file containing all filenames to be read. Because all three files that were created at the same time needed to be read to create a single case just one filename for each time period was read in. This name would then be processed to create the filename of the other two files to be read. The three files were then read in using the FILEVAR option to create the case in the data set. The FILEVAR option defines a variable whose name you supply and whose change in value causes the INFILE statement to close the current input file and open a new one. All the examples in the SAS manuals show just one file being read each time the data step loops, there is no documentation on how to read multiple files during each loop. The solution to doing this proved simple enough; each file to be read during the loop needed to have a different variable specified on each FILEVAR option.

```

/*****/
/*          System statistical summation          */
/*                                                    */
/* Name: STATSUM                                     */
/* Summary: System statistics are gathered each hour. Three files */
/*          are created each time in a particular subdirectory. */
/*          This program reads a file created by the following */
/*          VMS command: DIR DISK*./COLUMNS=1/OUT=Z.LIS */
/*          The file Z.LIS contains all the type 1 filenames, types */
/*          2 and 3 can be created from this. The dataset created */
/*          from this is used to read all the data files into a */
/*          SAS data set. */
/*                                                    */
/*****/

```

/* This option eliminates all notes on the log */

options nonotes;

/* Read the data file containing the filenames of the data to be read */

```

data gsas.statfile;
infile 'disk$user01:[census.sysstat1]z.lis' pad;
input fname $25.;
length ftype $4.;
ftype = substr (fname,1,4);
if ftype not = 'DISK' then delete;
drop ftype;
run;

```

/* Input the previous data set and create filenames to be read */

```

data gsas.sysstat3;
length pre1 $4. pre2 $5. pre3 $10. ;
set gsas.statfile ;
prefix=scan(fname,1,'.');
```

```

len=length(prefix);
pre1=substr(prefix,1,4);
pre2=substr(prefix,5,5);
pre3=substr(prefix,10);
dir='disk$user01:[census.sysstat1]';
diskfile=dir || 'DISK' || pre2 || pre3 || '.LIS';
userfile=dir || 'USER' || pre2 || pre3 || '.LIS';
memfile= dir || 'MEM' || pre2 || pre3 || '.LIS';

/* Read in the 3 data files created each hour */

infile ifile filevar=diskfile pad;
input
/// dev1 $ 1-7 free1 63-70 trans1 73-76 / dev2 $ 1-7 free2 63-70 trans2 73-76
/ dev3 $ 1-7 free3 63-70 trans3 73-76 / dev4 $ 1-7 free4 63-70 trans4 73-76 ;

length nuser $4. nproc $4.;
infile jfile filevar=userfile pad;
input / nline $80.;
nuser = scan(nline, 6, ',');
nproc = scan(nline, 11, ',');

infile lfile filevar=memfile pad;
input // totppage 36-41 freepage 47-52 usepage 59-64
///// tsmpack 36-41 freesmpk 47-52 usesmpk 59-64
/ tiopack 36-41 freeiopk 47-52 useiopk 59-64
/ tlgpak 36-41 freeigpk 47-52 useigpk 59-64
// tnpdymem 35-41 fnpdymem 46-52 unpdymem 58-64
/ tpdymem 35-41 fpdymem 46-52 updymem 58-64;

/* The IF is needed because the leading zero is missing from single */
/* digit dates. */

if len=18 then pre3='0' || pre3;
day=substr(pre3,1,2);
hour=substr(pre3,3,2);
minute=substr(pre3,5,2);
second=substr(pre3,7,2);

drop pre3 dir prefix len pre1 fname nline;
run;

```

One other feature of SAS proved to be annoying while testing the SAS code. Each time a file is opened and records read in, three Note lines are written to the SAS log noting the filename and number of records read. When testing with a large number of files or actually running the procedure this can result in a very large log file. Once the code was debugged I specified the NONOTES system option to eliminate the large number of notes written to the log. Just one warning on doing this. The NONOTES

option eliminates ALL note records from all sources from the log. Some of these are needed to spot bugs in the code, so do not specify this unless you are sure the program works perfectly.

The SAS program that creates the SAS data set will run no matter how many files are in the directory. You would probably want to dedicate a separate directory for keeping these files. This program is only run sporadically to create a data set that is then merged with a master data set using the following code.

```
/******  
/*  
/*      Merge new data into master file  
/*  
/* NAME: SASCOMB  
/* SUMMARY: Combine the new system statistics with that  
/*      previously gathered. To make this routine automatic  
/*      the newly created merged file is then renamed to the  
/*      old filename after it has been deleted.  
/*  
/******
```

```
data gsas.sysstat1;  
set gsas.sysstat gsas.sysstat3;  
run;  
proc datasets library=gsas;  
delete sysstat;  
run;  
proc datasets library=gsas;  
change sysstat1=sysstat;  
run;
```

The individual ASCII files are then deleted. Various reports and graphs are then generated from the data to show system usage.

Important References:

On the FILEVAR option:

SAS Language Reference, Version 6, First Edition. Pages 381-382, 387.

On the NONOTES System option:

SAS Language Reference, Version 6, First Edition. Page 765.