

SAS Macros in the Land of "X" (the Operating System)

Harmon L. Jolley

Healthsource Provident Administrators, Inc.

Abstract

Our company has implemented a SAS® data warehouse which resides on a 7-node, 500 GB IBM® SP™ server. Over 100 users (data analysts and systems personnel) connect to the server via SAS/CONNECT® and TCP/IP from their OS/2® or Windows® workstations. To assist in managing the computing environment, we access operating system (AIX®) data from within the SAS System using the SAS Macro Facility and SAS interfaces to the operating system. Although the illustrations are based on the AIX operating system, the concepts apply to other platforms because of the portability of the SAS System.

Managing the Computing Environment

Storage

Our SAS data warehouse and SP server were implemented as a replacement for a mainframe-based data analysis/reporting environment. One of the strengths of the SAS System is that it allows users to create subsets of data from the warehouse for further analysis, and file systems were allocated on the SP for that purpose. Prior to the SP system, end-users were limited by the cost and complexity of mainframe storage, or by the available space on LAN drives. Although the SP provided many gigabytes of storage without mainframe cost, we still needed ways to analyze how the space was being used and manage that resource.

System Performance

Our objective with the SP System was that performance would be at least comparable to the same work done on the mainframe. Because of this, as well as the need to track the usage of the system, we needed a way to capture data relating to system performance.

Transitioning End-Users to the AIX System

The AIX operating system contains many commands, the wealth of which can intimidate end-users. At the

same time, there are basic functions such as allocating directories which are fundamental for using the SAS System within AIX. To assist with operating system commands, we have provided access to them from within the SAS System.

The Basic Ingredients

The essential parts of the applications which we developed are:

- **SAS interface to AIX via "PIPE" Commands**
The SAS System allows operating system commands to be invoked and results received via the PIPE option of the FILENAME statement. For example:

`filename fileref PIPE "command";`
- **AIX Environment Variables**
These variables are used to exchange data between the SAS System and the operating system.
- **The SAS Macro Language**
SAS macros allow for complex program logic to be stored and then accessed by users, with only the needed parameters being specified.
- **SAS System autoexec.sas and config.sas files**
These files provide the startup information to the SAS System. In the case of shared environments such as SAS for AIX, these files also provide standardized settings.

Managing Storage

To provide end-users and systems personnel a means to monitor storage, we have developed the "%spacejam" SAS macro. It is so named because we knew from experience in the LAN environment that users would be in a jam if there was no available space to store datasets!

The user invokes the macro and specifies the name of the directory that is to be analyzed. The macro invokes the "ls" AIX command with the "R" (recursive) option to list files across subdirectories. The macro code is as follows:

```
%macro spacejam(dir);
```

```
filename lsir PIPE "ls -lR &dir";
data work.files(drop=one first);
attrib owner length=$8
       directry length=$78
       file length=$40
       bytes length=8
       month length=$3
       day length=$2
       time length=$5
       group length=$8
       permit length=$9
       first length=$1;
retain directry("&dir");
infile lsir pad missover;
input first $ @;
if first = '/' then do;
    input @1 directry $
    delete;
end;
if first ne '-' then delete;
input @1
    permit $
    one $
    owner $
    group $
    bytes
    month $ day $
    time $
    file $ ;
run;
```

```
proc sort data=work.files;
by owner bytes;
run;
```

```
proc sql;
create view work.temp as
select owner, sum(bytes) as bytes format=comma17.
from work.files
group by owner;
```

```
title 'Disk Space Usage by File Owner';
proc print data=work.temp noobs uniform;
sum _numeric_;
run;
%mend spacejam;
```

A report is automatically produced by the macro which ranks storage usage by user. This allows the system administrator or individual users to identify the users who are consuming most of the space, and ask them to pursue some file cleanup. In addition, a dataset called work.files is created by the macro, and contains detail about each file under the specified directory.

Managing System Performance

To provide analysis of levels of system performance, as well as reports to management on usage of the system, we have developed an application which parses the contents of SAS Log window output. The SAS System config.sas file contains a command which causes the contents of the SAS Log window to be written to a file:

```
-altlog /path/filename
```

In our config.sas file, *filename* includes an environment variable containing system date and time which allows the file names to be generated uniquely for each user's SAS session.

The SAS System autoexec.sas file invokes the AIX command "vmstat" in order to capture operating environment statistics such as available memory and CPU wait time:

```
vmstat | grep -v avm | grep -v memory | grep -v '-'
```

The files from all interactive and batch SAS sessions are parsed at the end of each day, and data captured. The following variables are captured in the system performance statistics dataset.

Variable	Label
CLKTIME	Clock time
CPUTIME	CPU time
FREEMEM	Free memory
HOSTNAME	Host name
IDLECPU	Idle CPU %
LGINDATE	login date
LGINTIME	login time
NUMUSERS	Number of users on system
PARSELOG	Path and file name of log file parsed
PROCNAME	Name of SAS procedure run
QUERY	'Y' if SAS SQL Query Window used
SASBATCH	'Y' if procedure run in batch
SYSCPU	System CPU %
USERID	User ID
USRCPU	User CPU %
WAITCPU	Wait CPU %

Transitioning End-Users to AIX

To allow end-users to submit AIX commands from within the SAS System, we have developed the %aix SAS macro. The user simply specifies any valid AIX command as a parameter to the %aix macro:

```
%aix(AIX command);
```

For instance, a user may see a list of files within a directory by specifying

```
%aix(ls -l /directory/subdirrectory);
```

The output is displayed in the SAS Log Window.

To save users keystrokes and hassles of submitting "libname" statements, we have included most of the standard production data libraries in the SAS System autoexec.sas file. However, this has not always been possible, particularly if the directory path must vary according to the segment of data requested. To address this, we developed a SAS macro which only requires that the user know the account number of the customer needing a report:

```
%macro acctlib(custno);  
libname claims "/sas/data/&custno" access=readonly;  
%mend acctlib;
```

```
%acctlib(A1234);
```

Summary

The SAS System offers a very powerful interface to the operating system on the various platforms on which it is deployed. The data and functionality that we have been able to provide have helped us not only to move into the new data warehouse environment on our SP server, but also to demonstrate its value.

Credits and References

Working with me in the development of these SAS macros and AIX routines were:

Charles Dibiazue
Blake Hoskinson

If you have any questions or comments, please contact:

Harmon Jolley
Healthsource Provident Administrators, Inc.
Two Fountain Square
Chattanooga, TN 37402
phone: (423) 755-8282 e-mail: jolleyh@hlthsrc.com

SAS and SAS/CONNECT are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.

IBM, OS/2, AIX, and SP are registered trademarks of International Business Machines Corporation.

Windows is a registered trademark of Microsoft, Inc.