TRICKS AND TIPS FOR SUPPORTING THE SAS® SYSTEM UNDER VMS™

Thomas O. Martin II, Alcoa Laboratories

Abstract
This paper describes support techniques used in the Alcoa Technical Center VAX/VMS™ environment for the SAS System. SAS/AFTM, the SAS™ product for developing menu driven applications, was used to develop custom menu systems with an easy to use interface. Online VMS help has been provided for SAS software describing site specific information as well as the SAS HELP facility. A VMS bulletin board is available for user comments and the dissemination of information. In addition, a method for monitoring the usage of SAS software has been developed.

Introduction
Alcoa Technical Center (ATC) is the Research and Development headquarters of the Aluminum Company of America. The facility occupies 2300 acres northeast of Pittsburgh, Pa. The various divisions encompass Product Engineering, Computing, Chemistry, Metallurgy, Packaging, and Manufacturing processes. Currently the laboratory staffs more than 1200 engineers, scientists, and technicians who transform corporate goals into quality products and new technology.

After it was determined that our existing, in-house developed statistical software could not meet the current needs of our statisticians and was lacking in flexible graphics capability, the first SAS products were installed in 1985. We currently have SAS/GRAPH™, SAS/AF™, SAS/FS™, SAS/IML™, SAS/ETS™, SAS/CPE™ and SAS/QC™ installed on two VAX 8650s in a cluster. We have also licensed 50 copies of the PC SAS System. SAS software has provided the computer users at the laboratory with an integrated system for data analysis, statistics and graphics. The objective of this paper is to describe the methods and tools that are in place to support the SAS software products at Alcoa Technical Center. This paper was prepared for presentation at the SAS Users Group International 14, San Francisco, California.

End-User Computing Environment
The computing environment at the laboratory supports end-user computing by providing tools and computing resources that allow the user to become self-sufficient. The engineers and scientists have terminals, IBM personal computers, compatibles and APPLE Macintosh computers to access the distributed computing environment. Our goal is to provide easy-to-use interfaces for packages and easy movement of data between packages and computers. With the widespread use of desktop computers with easy to use software, the challenge is to make powerful software on the VAX also easy-to-use.

SAS software is used at the laboratory as a part of process modeling systems, analysis of experiments, post processing of data from database systems, and quality control. One very effective use is rapid prototyping. We have many projects that are relatively short-lived. Therefore, software to support these projects must be developed quickly. We have made significant impact on project schedules with the ability to quickly create data entry screens with a menu driven front-end. This ability has also allowed us to develop prototypes that show the feasibility of a software system before we begin a larger development effort. In many cases, the prototype becomes the basis for the production system.

SAS Courses
Adequate training is necessary to introduce new users to the SAS System. Courses in SAS software are offered three times a year. We have developed and teach a three day course covering the data step programming, basic procedures, and SAS/GRAPH. A more detailed SAS/GRAPH course is taught on-site by SAS Institute education personnel. We have also held courses in various areas of statistics using SAS software. One of the common complaints when a user's initially looks at SAS software, is the intimidating volume of manuals. The courses have given users enough knowledge to access the manuals for reference instead of using them to learn the package.

A one-day advanced course, primarily designed to teach efficiency techniques, has been added this year. This course will be in a workshop format responding to the needs of the attendees. The participants will be consulted prior to the course to see if they have any particular areas they would like to learn more about.

The other portions of the package such as SAS/ETS, SAS/IML, SAS/FS, and SAS/AF are not taught to the general user community specifically. Often these procedures are used to develop systems for the user by the applications or statistics support group.

Menu System
During the study which resulted in ATC's purchase of the SAS System, it was determined that there was a significant user population that needs the capabilities of the products from SAS Institute but does not want to learn the complexities of a new language. At that time we decided to develop a menu system, which would give the casual user the analysis tools needed but with an interface that handled the details of the SAS language. After studying the Sample Menu System developed by SAS Institute using SAS/AF, we decided that it was a good framework upon which to build the system we wanted.

The current Menu System is considerably larger than the base from which it was built. Extensive enhancements to the statistical, graphical, and data management capabilities were developed and are now available. Areas where the advanced users could execute their own SAS commands without leaving the menu system are provided. A utilities menu gives the user the power to change system options, execute PROC SORT, and have access to VMS. This UTILITY menu appears below:
Utility Menu
Select Option ==>
Select one of the options and press ENTER
1 Modify System Options
2 List specified VMS directory
3 Enter VMS commands
4 Sort Utility
5 Enter SAS commands directly

The MODIFY SYSTEM OPTIONS allows the user to change default menu system options:

Set System Options
Command ==>
Press END to submit or CANCEL to
return to previous screen
Enter an X or a number to change
the default options.
NOTE: if both Yes and No are
chosen, No is assumed
Routing SAS Output
Yes No
Log to file MENUPROF.LOG? or X
Log to Screen ? or X
Output Options
Linesize(80): 80
Pagesize(24): 24
Print to file MENUPROF.LIS? or X
Print Output on screen? X or _
Center Output on page? _ or X
Today's Date in heading? _ or X

The menu system helps some users learn to write native SAS programs. Many users save the SAS programs, generated by the SASAF program screens, to disk and then modify them for their specific needs.

One of the most difficult problems during the development of the menu system was deciding what functions and options should be included on the menus and program screens. We were positioning SAS Software to replace another software package that had been in use at Alcoa for many years. The old software needed considerable updating in order for it to meet current needs. A functional requirements document was developed as part of the study which resulted in the purchase of the SAS products. It was determined that the menu system should exceed the functional requirements identified in that document. The menu system, based on the capabilities of SAS Software, has given us a tool that can be expanded as the needs of the user community grow. We plan enhancements this year to most areas, including a method to exchange data with other software such as RS/1, a laboratory data handling package, and INGRES, a relational database system. We plan to provide a simple method to transfer data between the Menu System and personal computers.

Usage Monitoring
One of the other needs we had was to monitor SAS software usage. A tracking system was developed that counts each execution of SAS software. We track the use of the Menu System separately from the execution of the SAS System. SAS software is executed from a command procedure that contains executions of two other command procedures, USAGE1 and USAGE2. USAGE1.COM gets the current value of process parameters being monitored. These values are passed to USAGE2.COM as VMS global symbols. USAGE2.COM is placed after the invocation of SAS in the command procedure. It gets the final values for these parameters and subtracts the initial values to get the values for the current execution of the package being monitored. These values are then written to a specific log file for each package. The parameters collected are SYS$LOGIN, CPU time, Buffered IO, Direct IO, and page faults.

The usage procedures are generalized and are used to track all of our purchased software products. The command procedures used to capture the data are shown in Figures 1a and 1b.

Usage reports are printed at the end of each month summarizing each parameter. The most important reports produced are the Unique Users Report and a report tabulating the overall number of executions of the package. These reports enable us to track the growth of the use of the package with respect to new users and overall use. Graphs are produced that complement the reports (Figures 3 and 4). These reports and graphs can be compared to system monitoring data to help see the impact on the use of the software on CPU utilization. We merge the data with another accounting file that contains information about the user's division, which allows us to produce graphs and reports by division.

Using these statistics, we were able to intelligently purchase the license for the second clustered 8850 CPU. The generalized routines allow us to manage CPU loading; allocate costs to divisions and evaluate whether more software licenses should be purchased or discontinued.

The main reporting program, USAGE.SAS in Figure 2a, sets up macro variables used by the included programs: REP1, which reads in the data, and REP2, which produces the reports. REP1 is shown in Figure 2b and REP2 is shown in Figure 2c.

Bulletin Board System
Another tool to assist in the support of SAS Software at Alcoa Technical Center is a bulletin board system running under VMS. The VMS Bulletin Board system was developed by the Mellon Institute Computer
Engineering Center. The system supports multiple bulletin boards, each operating independently of the others, and permits users access to the ones in which they are interested. We have bulletin boards for SAS Engineering Center. The system supports multiple bulletin boards, each operating independently of the others, and permits users access to the ones in which they are interested. We have bulletin boards for SAS software, Apple Macintosh, and IBM PC users. There is a bulletin board for items of general interest. The SAS software bulletin board in particular, provides a user forum for comments and a place where we can put tips and support messages. When a new version of the SAS System has been installed, information about the update is posted on the bulletin board. This is a place where site specific changes can be made public. Messages can be posted by VMS users to any of the bulletin boards.

SAS Help files
Help files play an integral part in the support of SAS software. We have implemented extensions to the standard VMS help files to include ATC site-specific software packages. We created help files for the SAS modules and included site-specific information. Users wanted to be able to have help available to them outside of the SAS interactive environment. The SAS help menu appears below:

<table>
<thead>
<tr>
<th>Additional Information available:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF BASICS ETS FSP GRAPH IML QC</td>
</tr>
<tr>
<td>PLOTTERS MANUALS MENU</td>
</tr>
<tr>
<td>MODES_OF_EXECUTION COMMANDS</td>
</tr>
</tbody>
</table>

SAS User Group
We occasionally have local SAS user group meetings at the lab. The purpose of the meetings is to provide another method by which we can discuss advanced topics, enhancements provided in new versions, and demonstrate capabilities of the products. Participation in the local Pittsburgh SAS User Group has helped considerably by providing a source for ideas, techniques, and presentations that can be used in user consulting as well as SAS application development.

Common Criticisms
The criticisms of SAS that we hear are the most common ones. The first is that documentation is thought to be overwhelming. The first time SAS user is usually armed with the SAS Introductory Guide and the SAS User's Guide Basics. The SAS Introductory Guide is too limiting and the Basics Guide is more like a reference manual than a learning guide for the SAS System. The SAS System for Elementary Statistical Analysis is an excellent example of the style of manual the typical beginning user of SAS can benefit from immediately. The guide is well written and provides the correct balance between a tutorial and a reference guide and even stays open on your desk.

The second common criticism is the user interface. Specifically, users seem to want the flexibility of the SAS System but with the interface of a turnkey system. Some systems are available today where the user is prompted for information, the user responds and the system provides an easy to read result. This is an excellent selling point for a software system. Many times, however, after these systems are in place for a while, the user wants to perform a function not specifically provided with the software. The work being done by the SAS Institute with its prototypes and the ASSIST™ system in SAS 6.03 is a step in the right direction.

Future
We predict desktop computers will play a much broader role in the near future. With larger, faster disk drives, more memory capacity and faster processors, the desktop computer is not just the simple productivity tool it once was. Desktop computers provide a wealth of processing power. One of the strong points of their software is the excellent user interface provided by some commonly used packages. We would like to work towards an environment where we have software that provides cooperative processing between desktop computers, including IBM PC compatibles and Apple Macintoshes, and our VMS and UNIX environments. The micro-mainframe link of PC SAS is an excellent example. We would like to have an even closer link. For example, an application could be running on the mainframe and display SAS/AF screens on the desktop computer not using a terminal emulator but a closer link allowing the user to click on a box using a mouse when filling in an action field or positioning the cursor to fill in a blank. The LOG and PRINT windows should be scrollable at the PC. This capability should be available without having the PC/SAS product on the desktop computer. Possibly the support of the DecWindows product will get us to this goal.

Wish List
A SQL interface to the SAS System would be helpful, in fact necessary, as more software adopts the SQL standard. There would be obvious benefits to this if the user is already experienced with relational databases. This would provide a common query language for SAS datasets and relational databases. The new desktop computer tools such as SOL window packages would then be able to interface to SAS datasets as well.

User documentation written in the style of the SAS Series in Statistical Applications would be an excellent addition to the reference material currently available. The user needs the reference manuals currently available but also needs an easy to read guide to the SAS data step processing.

Conclusions
Support of a product as diverse as the SAS System presents many challenges to support personnel. The facilities described in this paper are intended to assist the SAS support staff to improve the level of support they can offer their users. The tools and methods we have in place help us identify the users of the product and supply tools to enable them to use the product effectively.

Acknowledgements
The author wishes to thank SAS Institute for the initial development of the Sample Menu System software, Blazek, Alcoa Laboratories, for her work on the Alcoa Menu System.
The author may be contacted at:
Alcoa Technical Center
RT 780,
Alcoa Center Pa. 15069

References

*SAS, SAS/GRAPH, SAS/FSP, SAS/AF, SAS/ETS, SAS/QC, SAS/IML, and SAS/CPE are registered trademarks of SAS Institute Inc, NC, USA.

*DEC, VMS, and VAX are registered trademarks of Digital Equipment Corporation, Maynard, Massachusetts.

```sas
$ VER = $VERIFY()
$ SET NOVERIFY
$!----------------------------------------------------------
$!*Get start values
$!----------------------------------------------------------
$ DATE1 = $TIME()
$ TIME1 = $CVTIME(DATE1,"TIME")
$!----------------------------------------------------------
$!* CPU time
$!* BUFI0 = Buffered I/O operations
$!* DIRIO = Direct I/O operations
$!* PAGEFLTS = Page faults
$!----------------------------------------------------------
$ CPUTM1 = $GETJPI("","CPUTM")
$ BUFI01 = $GETJPI("","BUFI0")
$ DIRIO1 = $GETJPI("","DIRIO")
$ PAGEFLT1 = $GETJPI("","PAGEFLTS")
$ IF VER .EQ. 1 THEN SET VERIFY
$ EXIT
$

Figure 1a - USAGE1.COM

$! The current process name is passed
$! to this DCL command procedure as a
$! positional argument.
$! This records the current process
$! name (from PL) and time, date,
$! CPU time, BUFFERED I/O,
$! DIRECT I/O, and PAGE FAULTS.
$! SET NOVERIFY
$!----------------------------------------------------------
$!* Define file to record usage data.
$! FI is process name, such as SAS
$! FILE = "DISK2:USAGE1/'FI'_LOG.DAT"
$!
$!* Obtain root directory name.
$!
```

```sas
$ D = $STRlm("SYS$LOGIN")
$ A = $LOCATE("[",D)+1
$ B = $LENGTH(D) - A - 1
$ NAME = $EXTRACT(A,B,D)
$!

$** Record current date and time
$!----------------------------------------------------------
$ DATE2 = $TIME()
$ TIME2 = $CVTIME(DATE2,"TIME")
$!----------------------------------------------------------
$** Record parameters after execution
$!----------------------------------------------------------
$ CPUTM2 = $GETJPI("","CPUTM")
$ BUFI02 = $GETJPI("","BUFI0")
$ DIRIO2 = $GETJPI("","DIRIO")
$ PAGEFLT2 = $GETJPI("","PAGEFLTS")
$!----------------------------------------------------------
$** Take differences
$!----------------------------------------------------------
$ CPUTMT = CPUTM2 - CPUTM1
$ BUFIOT = BUFI02 - BUFI01
$ DIRIOT = DIRIO2 - DIRIO1
$ PAGEFOT = PAGEFLT2 - PAGEFLT1
$!----------------------------------------------------------
$** Check if usage file already exists
$! if not then create file.
$!----------------------------------------------------------
$ IF $SEARCH("~~UFILE'",).NBS ..... THEN
$ GOTO FILE EXISTS
$ OPEN/APPEND/ERROR=ERROR_MESSAGES USAGE_FILE
$ WRITE_DATA
$ FILE EXISTS:
$ OPEN/APPEND/ERROR=ERROR_MESSAGES
$ USAGE_FILE 'UFILE'
$!
$!----------------------------------------------------------
$** Write usage values to usage file
$!----------------------------------------------------------
$ WRITE USAGE_FILE PI, "", NAME,".. ,TIME1,".. ,CPUTM1," ..,BUFI01, ..,DIRIO1, ..,PAGEF1, ..
$ IF VER .EQ. 1 THEN SET VERIFY
$ EXIT
$!

$!----------------------------------------------------------
$** Write error messages
$!----------------------------------------------------------
$ ERROR MESSAGES:
$ WRITE SYS$OUTPUT "Error opening "'UFILE'
$ IF VER .EQ. 1 THEN SET VERIFY
$ EXIT
$

Figure 1b - USAGE2.COM
```
OPTIONS LTYFE;
%LET SUB4=XDIV;
%LET SUB1=SAS;
%LET SUB11=F30;
%LET SUB12=U30;
%LET SUB2=OEV$CLUSTER2:(USAGE)SAS LOG.DAT;
%LET SUB3=DEV$CLUSTER2:{U5AGEli - 
%-INC REP1;
%INC REP2;
RUN;

Figure 2a - USAGE.SAS

OPTIONS MOLT YFE CENTER LS=132;
FILENAME IN ’&SOO2’;
LIBNAME xxx ”&SUB3’;
LIBNAHE PERM’ ()’
/* XDIV CONTAINS A PROC FORMAT MAPPING
DIVISION NUMBERS TO DIVISION NAMES */
%INC XDIV;

DATA USAGE;
LENGTH USER $ 15;
INFILE IN;
INPUT PACKAGE $ USER$ START: TIMElO.
DATE: DATEIO# STOP: TIMElO.
CPU BIO 010 PAG
DURATION=STOP-START;
IF DURATION<0 THEN
DURATION=DURATION+24*3600;
CPU = CPU / 100. ;
ACC = 1 ;
DURATION = DURATION/60 ;
ELAP = DURATION ;
MON =MONTH(DATE) ;
YEAR =YEAR(DATE) ;
MONTH =MDY(MON,01,YEAR) ;
NOW =TODAY() ;
FORMAT MONTH MONYY6. ;
FORMAT START TIME10. ;
FORMAT STOP TIME10. ;
FORMAT DURATION 'lIMBIO. ;
FORMAT DATE DATE9. ;
IF YEAR=YEAR (NOW) AND
MON=MONTH (NOW) AND
DAY (NOW)<28 THEN DELETE;
RUN;

Figure 2b - REP1.SAS

DATA NAME; SET xxx.$SUB4;
PROC SORT DATA=USAGE; BY USER ;
PROC SORT DATA=NAME; BY USER ;

DATA PERM.MERGED;
MERGE USAGE(IN=INU) NAME(IN=INN) ; BY USER;
IF INU ; IF NOT INN THEN DIV=’?’;
RUN;
PROC DATASETS;
DELETE USAGE;
RUN;
PROC SORT DATA=PERM.MERGED; BY USER DATE;

Figure 2c - REP2.SAS

PROC SORT DATA=NAME; BY USER;
PROC SORT DATA=USAGE; BY USER ;

DATA USAGE ; SET PERM.MERGED; BY USER DATE;
KEEP USER USES LAST_USE DIV;
IF FIRST.USER THEN USES = 0;
USES +1;
IF LAST.USER THEN DO;
LAST_USE = DATE;
OUTPUT;
END;
FORMAT LAST_USE DATES. ;
RUN;

PROC PRINT;
TITLE ”&SUB1 USERS - YEAR TO DATE - SORTED BY
ACCESSSES”;
FORMAT DIV $DNAME. ;
PROC SORT DATA=PERM.MERGED; BY MONTH DIV;

PROC TABULATE DATA=PERM.MERGED;
CLASS MONTH DIV :
VAR ACC DURATION CPU BIO DIO PAG;
LABEL DIV=’DIVISION’
ACC=’ACCESSSES’
DURATION=’ELAPSED (MIN)’
CPU=’CPU (SEC)’
BIO=’BUFFERED I/O’
DIO=’DIRECT I/O’
PAG=’PAGE FAULTS’ ;
TABLES (MONTH ALL) * (DIV ALL) ;
(ACC*F=8. DURATION*F=9. CPU*F=10.2
BIO*F=9. DIO*F=9. PAG*F=9.);
ESTLABEL ALL=’TOTAL’ SUM =’’;
TITLE ”&SUB1 USAGE PER MONTH AND DIVISION”;
FORMAT DIV $DNAME. ;
PROC CHART DATA=PERM.MERGED
VBAR MONTH / DISCRETE TYPE=SUM SUMVAR=ACC;
TITLE ”&SUB1 USAGE PER MONTH”;
TITLE2 ”ACCESSSES”;
FORMAT DIV $DNAME. ;
PROC MEANS NOPRINT DATA=PERM.MERGED SUM;
BY MONTH DIV;
VAR ACC ELAP CPU BIO DIO PAG;
ID PACKAGE USER;
OUTPUT OUT=XXX. &SUB11 SUM=ACC ELAP CPU BIO
DIO PAG;
PROC SORT DATA=PERM.MERGED; BY MONTH USER;

PROC CHART DATA=PERM.MERGED
VBAR MONTH / DISCRETE TYPE=SUM SUMVAR=ACC;
TITLE ”&SUB1 USAGE PER MONTH”;
TITLE2 ”ACCESSSES”;
FORMAT DIV $DNAME. ;
PROC MEANS NOPRINT DATA=PERM.MERGED SUM;
BY MONTH DIV;
VAR ACC ELAP CPU BIO DIO PAG;
ID PACKAGE USER;
OUTPUT OUT=XXX. &SUB11 SUM=ACC ELAP CPU BIO
DIO PAG;
PROC SORT DATA=PERM.MERGED; BY MONTH USER;

DATA XXX. &SUB12; SET PERM.MERGED; BY MONTH USER;
KEEP MONTH USER PACKAGE;
IF FIRST.USER THEN OUTPUT;

DATA UNIQUE; SET PERM.MERGED; BY MONTH USER;
KEEP MONTH UNIQ;
IF FIRST.MONTH THEN UNIQ = 0;
IF FIRST.USER THEN UNIQ +1;
IF LAST.MONTH THEN DO;
OUTPUT;
UNIQ = 0;
END;
PROC CHART DATA=UNIQUE; VBAR MONTH / DISCRETE
FREQ=UNIQ;
TITLE ”&SUB1 USAGE PER MONTH”;
TITLE2 ”NUMBER OF UNIQUE USERS”;

Figure 2c - REP2.SAS

511