INTERACTIVE TECHNIQUES TUTORIAL

J.D. Young, Texas Instruments, Inc.
Cynthia Deitz, Texas Instruments, Inc.

PROBLEM STATEMENT

This tutorial topic illustrates techniques in interactive processing that can greatly aid productivity. SAS running under CMS offers the user much flexibility and power, which if not used properly can be very detrimental. It is the intent of this tutorial to present information that will enable a SAS/CMS user to become more proficient. The following examples range from quite simple to relatively complex.

ENVIRONMENT INFORMATION

It is informative during an interactive session to determine what resources SAS is using. This can be done by issuing the SAS command PROC OPTIONS SHORT; as follows:

```
  1? proc options short; run;
SYSTEM PARAMETERS AND OPTIONS

BAUD=1200 NOBLDLTABLE BLKSIZE=2048 BUFNO=1 CAPS C=MAX C60 NOCENTER NOCHARCODE
CHKPT NODATE DEVADDR= DEVICE= DISK=SYSDA NODQUOTE NODUMP NODYNALLOC NOERRORABEND
ERRORS=20 FIRSTOBS=1 GEN=5 FILSZ GRAPHICS INCLUDE INVALIDDATA=. LABEL
_LAST_=_NULL_ LEAVE=60535 LOG=FT11F001 NOLOWERCASE LS=132 NOMACROGEN MISSING='.'
MODECHARS='?>*' NONEWS NOTES NONUMBER OBS=MAX OFFLINE=0.0040 ONLINE=0.0120
NOOPLIST NOOVP P=MAX PARM='' PARMCARDS=FT15F001 PROCSIZE=16776192
PROMPTCHARS='110A010D09000000'X PS=24 REPLACE S=80 S2=S SEQ=8 SKIP=0 NOSNP
NOSNPPROG SORT=0 SORTDEV=SYSDA SORTLIB='SORTT' NOSORTLIST SORTSIZE=SIZE SOURCE
SOURCE2 NOSPOOL SYSIN=SYSIN SYSPARM='' TAPE=SYSSQ T=MAX TLS=80 TRANTAB=GTABCMS
UNITS=11 12 13 14 15 16 17 18 19 20 USER=WORK WORK=WORK NOWORKINIT

NOAUXDIR CMSSVC=255 CPSP ERASE NOFILCLR FORTG=''   LTYPE NAME='' NONULLEOF
PLIF=SPLILIB PLIO=PLILIB  PSEG=OFF PTYPE SASLIB='' SERIES='' SIODISK=A SSEG=OFF
NOTLOG TMSGLEV=ERRORS TXTLIB VIOBUF=5
```

Some important information contained above is:

1)Where the log files are being sent; in this case to FT11F001.
2)What I/O units SAS is using. This is important if it is necessary to call a FORTRAN program from within SAS, or any other software that might allocate I/O units. SAS has control of these units and will not relinquish this control until the SAS session has ended.
3)The LEAVE parameter indicates the memory left to CMS. This is important to know when it is necessary to execute CMS commands during the SAS session. For example, trying to edit a large dataset might cause a memory error.

This proc provides the opportunity to check option settings in a convenient manner without interrupting the current session.

Dataset information may also be acquired by using PROC DATASETS as follows:

```
28? proc datasets ddname=job; run;

                    CMS/SAS DATABASE FILETYPE=JOB, FILEMODE=A1
                    LIST OF DATA SETS BEFORE UPDATE OF DIRECTORY.
                    NAME             OBS PROT
                    TEST              20
                    TEST2             30   P
                    NOTE: NO CHANGE TO DIRECTORY.
```

In addition to indicating all datasets in the directory for the member "JOB", the datasets that are password protected are also flagged. In this case dataset JOB.TEST2 is password protected. Having this capability makes it much easier to keep track of datasets during an interactive session.

EXECUTING CMS COMMANDS FROM WITHIN SAS

It is sometimes convenient to be able to execute CMS commands from within SAS. One such command is FLIST. For example:

```
7? cms flist * * a;
```

```
 LVL 0 --- A 191  15000 BLKS 3370 R/W   1205 FILES 70% -- FILE    969 OF   1205
WATERLOO RELEASE2 A1                      F  1024     46     46  9/11/81 15:37
WATERLOO RELEASE3 A1                      F  1024     74     74  9/11/81 15:38
WATERLOO RELEASE5 A1                      F  1024     62     62  9/11/81 15:38
WATERLOO RELEASE6 A1                      F  1024     27     27  9/11/81 15:38
WATERLOO RELSP    A1                      F  1024     68     68  3/17/82 17:32
PROFILE  RSCS     A1                      F    80     18      2  6/19/81 15:26
CHARTY   SAS      A1                      F    80     76      6  1/21/81  0:04
CHKPLOT  SAS      A1                      F    80      2      1  8/14/81 17:34
DEMO1    SAS      A1                      F    80      7      1  2/11/81 23:12
DEMO2    SAS      A1                      F    80     14      2  2/11/81 23:52
EXTR     SAS      A1                      F    80      9      1  2/01/81 20:07
GSLIDE   SAS      A1                      F    80     11      1  1/27/82 18:50
G3D      SAS      A1                      F    80     12      1  1/27/82 18:26
G3D2     SAS      A1                      F    80     12      1  7/27/82 15:46
NEEDLE   SAS      A1                      F    80     12      1  7/27/82 16:01
PROG2    SAS      A1                      F    80     45      4  7/27/82 15:56
SAS01    SAS      A1                      F    80     26      3  2/07/81 18:04
SAS02    SAS      A1                      F    80     26      3  2/07/81 18:20
SAS03    SAS      A1                      F    80     27      3  2/07/81 18:36
TEACH    SAS      A1                      F    80     29      3  1/08/81 18:56
X        SAS      A1                      F    80     11      1  2/16/81 18:15
1=I 2=BRW 3=END 4=XED 5=SPL 6=/ST 7=SCB 8=SCF 9=/SD 10=P38 11=P66 12=CAN
```

From this screen it is possible to browse, edit, delete, copy, or rename files very easily. This is especially useful when the "A" disk is full and it is necessary to "find" some space.

Another CMS command to consider is Q DISK:

```
 17? cms q disk;
LABEL  CUU M  STAT CYL TYPE BLKSIZE    FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
YOU191 191 A   R/W   1 3350  1024        38       342-76        108        450
LICC9F 19F C   R/O  FB 3370  1024       412      8392-73       3108      11500
SAS191 204 E   R/O  FB 3370  1024       273      9536-95        464      10000
BETA91 205 F   R/O  FB 3370  1024        63      2827-63       1673       4500
DCS    19A O   R/O  FB 3370  1024       498     27745-73      10505      38250
SYSTEM 190 S   R/O  FB 3370  1024       229     16558-75       5586      22144
IBMPP  19E Y/S R/O 100 3350  1024       463     32019-71      12981      45000
```

This command is also useful in determining the amount of space left on a disk.

The CMS command FILEDEF can also be advantageous:

```
 14? cms filedef;
SYSIN    TERMINAL
FT11F001 TERMINAL
SYSOUT   TERMINAL
FT12F001 TERMINAL
FT13F001 DISK     SAS      SASPUNCH A1
LIBRARY  DISK     $SASLIB  TXTLIB   A1
SASTXTL  DISK     SASTXTL  TXTLIB   E1
SORTLIB  DUMMY
SORTWKOX DISK     SM023WRK SORTWK01 A1
WORK     DISK     #DIRMACR WORK     A1
FT15F001 DISK     $SASPC   PARMCARD A1
```

This provides a useful mechanism for determining what FILEDEFs have been declared. A note of warning: if SAS has established a FILEDEF for a file and another program is executed from within SAS which also tries to establish FILEDEFs with the same DDNAME, the program's FILEDEF will be ignored.

INTERFACING SAS, EXEC, FSP, AND DMS.

The following example is taken from a tool that uses CMS EXEC, Display Management System(DMS), SAS and SAS FSP. The tool was developed for a highly non-technical user and is a database management system. The database manager is a set of SAS programs. The user interfaces are EXEC and SAS FSP. The example shows that through the EXEC program, the user is entirely insulated from SAS and is not required to learn SAS in anyway.

The EXEC program and DMS panels follow. An explanation of selected portions of the EXEC follows the code and panels.

```
&CONTROL OFF
*************************************************************************
*                        REQUIREMENTS                                   *
*                                                                       *
*  THE FOLLOWING SCREENS ARE CALLED BY THIS EXEC:                       *
*                                                                       *
*  1) JOB. THIS IS THE FIRST AND PRIMARY SCREEN CALLED.                 *
*  2) JOBHLP. THIS IS THE HELP SCREEN THAT EXPANDS ON THE JOB SCREEN.   *
*                                                                       *
*     A) MODSCRN. THIS SCREEN PROVIDES A MENU FOR ACTIONS TO BE TAKEN   *
*                 IN THE MODIFICATIONS WORLD.(SEE REQ.S DOCUMENT)       *
*     B) MODHELP. THIS IS THE FIRST PAGE OF THE HELP SCREEN FOR A.      *
*     C) MODHELP2. THIS IS PAGE 2 OF THE HELP SCREEN FOR A.             *
*     D) QRSCRN. THIS SCREEN PROVIDES A MENU FOR ACTIONS TO BE TAKEN    *
*                IN THE QUERY WORLD. (SEE REQUIREMENTS DOCUMENT)        *
*     E) QRHLP. THIS IS THE HELP SCREEN FOR D.                          *
*                                                                       *
*  SCREENS A AND D INTERFACE WITH SAS.                                  *
*                                                                       *
*************************************************************************
*                                                                       *
*                          FLOW                                         *
*                                                                       *
*  THE PRIMARY MENU IS DISPLAYED. THE USER SELECTS FROM AMONG THE       *
* OPTIONS ON THIS MENU. USE OF AN INACTIVE KEY RESULTS IN THE DISPLAY   *
* OF AN ERROR MESSAGE AND A RETURN TO THE PRIMARY MENU. BASED ON THE    *
* USER'S SELECTION, ANOTHER MENU IS DISPLAYED, OR THE USER RETURNS TO   *
* CMS. POSSIBLE MENUS FROM THE PRIMARY SCREEN ARE:                      *
*    1) HELP                                                            *
*    2) MODIFICATIONS WORLD                                             *
*    3) QUERY WORLD                                                     *
*                                                                       *
*************************************************************************
* DISPLAY THE PRIMARY MENU. THE MENU USES ONLY PF KEYS.                 *
*************************************************************************
&BEGSTACK LIFO
ENDLIST
&END
EUDCMSX JOB
*=======================================================================*
* EXAMINE THE RESULT RETURNED BY THE USER. IF AN ERROR OCCURS           *
* IN CALLING THE DISPLAY, THE EXEC ABORTS WITH AN APPROPRIATE           *
* MESSAGE.                                                              *
*************************************************************************
&IF &RETCODE NE 0 &GOTO -DONE
*
* RSTATUS IS THE VARIABLE RETURNED BY THE SCREEN MANAGER.
*
* PF1 IS THE HELP PANEL
&IF &RSTATUS EQ PF1 &GOTO -HLP
*
* PF3 IS THE 'QUIT' OPTION. CONTROL IS RETURNED TO CMS.
&IF &RSTATUS EQ PF3 &GOTO -DONE
*
* PF4 IS THE MODIFICATIONS WORLD OF THE AUTOMATED FILE SYSTEM.
&IF &RSTATUS EQ PF4 &GOTO -MODS
*
* PF5 IS THE QUERIES WORLD OF THE AUTOMATED FILE SYSTEM.
&IF &RSTATUS EQ PF5 &GOTO -QUERIES

*
* UNRECOGNIZABLE INPUT. TRY AGAIN.
&SPACE 10
&BEGTYPE
          INPUT ERROR: USE ONLY ACTIVE PF KEYS
          HIT 'ENTER' TO RETURN TO MENU
&ENDTYPE
&READ
*
*  RETURN TO THE PRIMARY MENU
```

```
&GOTO TOP
*=====================================================================*
*                                                                     *
* THIS BLOCK CALLS THE HELP PANEL FOR THE PRIMARY JOB SCREEN.         *
* IF THERE IS AN ERROR IN CALLING THE SCREEN, THIS EXEC REPORTS THE   *
* ERROR AND EXITS THE EXEC.                                           *
***********************************************************************
-HLP
&BEGSTACK LIFO
ENDLIST
&END
EUDCMSX JOBHLP
&IF &RETCODE NE 0 &GOTO -DONE
* THE PRIMARY PANEL IS RECALLED.
&GOTO TOP
***********************************************************************
*                  MODIFICATIONS WORLD                                *
*                                                                     *
*  THIS IS THE MODIFICATIONS WORLD OF AFS. SEE REQUIREMENTS           *
* DOCUMENT FOR DETAILS.                                               *
*                                                                     *
*   WHEN THE MODS WORLD MENU IS DISPLAYED, THE USER HAS THE FOLLOWING*
* OPTIONS:                                                            *
*   1) SEE A HELP MENU                                                *
*   2) RETURN TO CMS                                                  *
*   3) RETURN TO THE PRIMARY MENU                                     *
*   4) CREATE A MODSET                                                *
*   5) UPDATE ANY OR ALL OF THE SAS DATA SETS.                        *
*   6) BROWSE THE MESSAGE FILE GENERATED BY THE UPDATE PROGRAM.       *
*                                                                     *
***********************************************************************
*
* IF THERE IS AN ERROR IN CALLING MODIFICATIONS WORLD SCREEN, THE
* EXEC ABORTS, WITH THE APPROPRIATE MESSAGE.
-MODS
&BEGSTACK LIFO
ENDLIST
&END
EUDCMSX MODSCRN
&IF &RETCODE NE 0 &GOTO -DONE
*
*  THE RESULT FROM THE PANEL IS ANALYZED.
*=====================================================================
* PF01 IS THE HELP SCREEN
&IF &RSTATUS EQ PF1 &GOTO -MODHLP
*
* PF02 IS THE POWER QUIT
&IF &RSTATUS EQ PF2 &GOTO -DONE
*
* PF03 RETURNS TO THE PRIMARY MENU

&IF &RSTATUS EQ PF3 &GOTO TOP
*
* PF04 ENTERS THE SAS FSP TOOL IN THE FSEDIT MODE.
&IF &RSTATUS EQ PF4 &GOTO -REQEDIT
*
* PF05 RUNS THE SAS MASTER UPDATE PROGRAM FOR REQUESTS FILE ONLY
&IF &RSTATUS EQ PF5 &GOTO -REQUPD
*
* PF06 PERMITS THE USER TO BROWSE A MESSAGE FILE CONTAINING RESULTS OF
* REQUEST FILE UPDATE.
&IF &RSTATUS EQ PF6 &GOTO -REQUPDL
*
* PF07 ENTERS THE SAS FSP TOOL IN THE FSEDIT MODE.
&IF &RSTATUS EQ PF7 &GOTO -EMPEDIT
*
* PF08 RUNS THE SAS MASTER UPDATE PROGRAM FOR EMPLOYEES FILE ONLY
&IF &RSTATUS EQ PF8 &GOTO -EMPUPD
```

```
*
* PF09 PERMITS THE USER TO BROWSE A MESSAGE FILE CONTAINING RESULTS OF
* EMPLOYEE FILE UPDATE.
&IF &RSTATUS EQ PF9 &GOTO -EMPUPDL
*
* PF10 RUNS THE SAS MASTER UPDATE PROGRAM FOR BOTH FILES
&IF &RSTATUS EQ PF10 &GOTO -ALLUPD
*
* PF11 PERMITS THE USER TO BROWSE A MESSAGE FILE CONTAINING RESULTS OF
* REQUEST FILE UPDATE.
&IF &RSTATUS EQ PF11 &GOTO -ALLUPDL
*
* ON ERROREOUS INPUT, REDISPLAY THIS SCREEN
&SPACE 10
&BEGTYPE
        INPUT ERROR: USE ONLY ACTIVE PF KEYS
        HIT 'ENTER' TO CONTINUE
&ENDTYPE
&READ
&GOTO -MODS
*=============================================================
*
* THESE ARE THE HELP SCREENS FOR THE MODIFICATIONS SCREEN
-MODHLP
&BEGSTACK LIFO
ENDLIST
&END
EUDCMSX MODHELP
&IF &RETCODE NE 0 &GOTO -DONE
&BEGSTACK LIFO
ENDLIST
&END
EUDCMSX MODHELP2
&IF &RETCODE NE 0 &GOTO -DONE
* THE MODIFICATIONS PANEL IS RECALLED.
&GOTO -MODS
*
*  THIS BLOCK INTERFACES WITH SAS.  THE RETURN FROM SAS IS EXAMINED.
* ON ERROR, THE EXEC PRINTS A MESSAGE AND RETURNS TO THE MODS WORLD.
*
-REQEDIT
CLRSCRN
&SPACE 10

&TYPE               PLEASE WAIT FOR PAGE TO APPEAR
&STACK HT
&BEGSTACK
PROC FSEDIT DATA=JOB.REQUPDF(READ=LIDO PROTECT=BITA)
SCREEN=SCREEN.REQUPDAT OPTION=1; RUN;
/*
&END
SAS (NAME REQMOD @DISK DATE)
&STACK RT
&IF &RETCODE NE 0 &GOTO -MODERR
*
* RETURN TO THE MODIFICATIONS SCREEN
&GOTO -MODS
*
*  THIS BLOCK RUNS THE SAS PROGRAM THAT UPDATES THE MASTER REQUESTS FILE
* WITH THE MODSET CREATED FOR THAT FILE.
-REQUPD
CLRSCRN
&SPACE 10
&BEGTYPE
              UPDATING MASTER REQUESTS FILE
              PLEASE WAIT
&ENDTYPE
```

```
&STACK HT
SAS UPDREQ (NAME UPDREQS LDISK PDISK
&STACK RT
&IF &RETCODE NE 0 &GOTO -UPDERR
&GOTO -MODS
*
*  THIS BLOCK PERMITS THE USER TO BROWSE THE MESSAGE FILE
* PRODUCED BY THE SAS PROGRAM THAT UPDATES THE REQUESTS FILE.
-REQUPDL
BROWSE RQUPDATE MESSAGES A
&IF &RETCODE NE 0 &GOTO -BROWERR
&GOTO -MODS
*
*  THIS BLOCK INTERFACES WITH SAS. ON ERROR THE EXEC PRINTS A
* MESSAGE AND RETURNS TO THE MODS WORLD.
-EMPEDIT
CLRSCRN
&SPACE 10
&TYPE                PLEASE WAIT FOR PAGE TO APPEAR
&STACK HT
&BEGSTACK
PROC FSEDIT DATA=JOB.EMPUPDF (READ=LKJH PROTECT=ZXCV)
SCREEN=SCREEN.EMPUPDAT OPTION=1; RUN;
/*
&END
SAS (NAME EMPMOD DATE PDISK)
&STACK RT
&IF &RETCODE NE 0 &GOTO -MODERR
*
* RETURN TO THE MODIFICATIONS SCREEN.
&GOTO -MODS
*
-EMPUPD
CLRSCRN
&SPACE 10
&BEGTYPE
                UPDATING MASTER EMPLOYEES FILE
                PLEASE WAIT
&ENDTYPE
&STACK HT
SAS UPDEMP (NAME UPDEMPS LDISK PDISK)
&STACK RT
&IF &RETCODE NE 0 &GOTO -UPDERR
&GOTO -MODS
*  THIS BLOCK PERMITS THE USER TO BROWSE THE MESSAGE FILE
* PRODUCED BY THE SAS PROGRAM IN EMPLOYEE UPDATE STEP.
-EMPUPDL
BROWSE EMUPDATE MESSAGES A
&IF &RETCODE NE 0 &GOTO -BROWERR
&GOTO -MODS
*
-ALLUPD
CLRSCRN
&SPACE 10
&BEGTYPE
                UPDATING MASTER FILES
                PLEASE WAIT
&ENDTYPE
&STACK HT
SAS UPDALL (NAME UPDALL LDISK PDISK)
&STACK RT
&IF &RETCODE NE 0 &GOTO -UPDERR
&GOTO -MODS
*
*  THIS BLOCK PERMITS THE USER TO BROWSE THE MESSAGE FILE
* PRODUCED BY THE SAS PROGRAM IN THE LAST STEP.
-ALLUPDL
BROWSE ALUPDATE MESSAGES A
&IF &RETCODE NE 0 &GOTO -BROWERR
&GOTO -MODS
```

```
*********************************************************************
*********  QUERY WORLD CODE OMITTED FOR BREVITY  *********************
*********************************************************************
*********************************************************************
*  ERROR MESSAGES.
-UPDERR
CLRSCRN
&SPACE 10
&BEGTYPE
THERE WAS AN ERROR IN UPDATING THE MASTER REQUESTS FILE.
HIT 'ENTER' TO RETURN TO MODIFICATIONS MENU.
&ENDTYPE
&RETCODE
&READ
&GOTO -MODS
*
-MODERR
CLRSCRN
&SPACE 10
&BEGTYPE
THERE WAS AN ERROR IN CREATING THE MODSET.
HIT 'ENTER' TO RETURN TO MODIFICATIONS MENU.
&ENDTYPE
&RETCODE
&READ
&GOTO -MODS
*

-BROWERR
CLRSCRN
&SPACE 10
&BEGTYPE
THERE WAS AN ERROR IN BROWSING THE MESSAGE FILE.
HIT 'ENTER' TO RETURN TO MODIFICATIONS MENU.
&ENDTYPE
&RETCODE
&READ
&GOTO -MODS
-DONE &EXIT &RETCODE
```

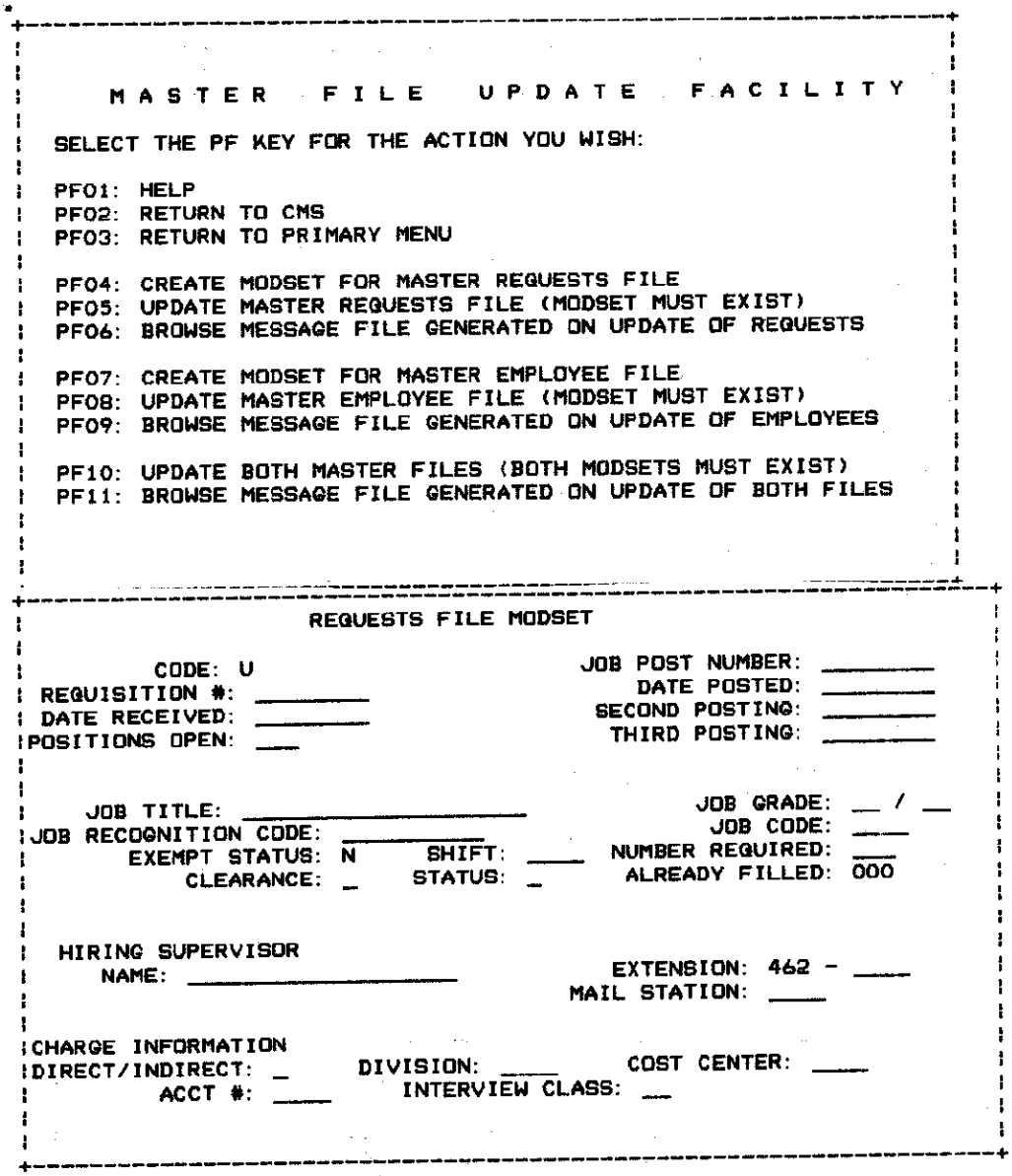
```
                    W E L C O M E   T O   T H E

                    JJJJJJJJ      OOO      BBBBBB
                       JJ       OO  OO     BB   BB
                       JJ       OO  OO     BBBBBB
                    JJ JJ       OO  OO     BB   BB
                     JJJ          OOO      BBBBBB

            I N T E R A C T I V E   F I L E   M A N A G E R

 SELECT THE PF KEY CORRESPONDING TO THE ACTION YOU WISH.

  PF01:  HELP
  PF03:  QUIT (LEAVE JOB FILE MANAGER)
  PF04:  MAKE MODIFICATIONS TO MASTER FILE
  PF05:  QUERY MASTER FILE
```



1 Shows the EXEC-DMS interface. The panels themselves are developed using the CMS PANEL tool.
2 Shows the EXEC-SAS FSP interface. Since it takes several seconds for the FSP panel to appear,
3 was inserted(Blank screens terrify users)
4 Halts transmission to the screen. This simply eliminates echo of commands entered by the EXEC or read from the console stack. 5 The FSP FSEDIT invocation is placed FIFO on the console stack.
6 Next, the exit SAS CMS command is stacked
7 Now, SAS is invoked, with a log to assist the consultant.

What the user sees is the MASTER FILE UPDATE FACILITY panel. If he then presses PF04 he will get the message: Please wait for screen to appear and then the FSP screen: REQUESTS FILE MODSET. When he has finished with this screen, he presses PF03 which will return him to the panel from which FSP was called, the MASTER FILE UPDATE FACILITY panel.