```
   CCCCCCCCCCC  LLL              IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC  LLL              IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC  LLL              IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

FILEID**SMOMSGUTL

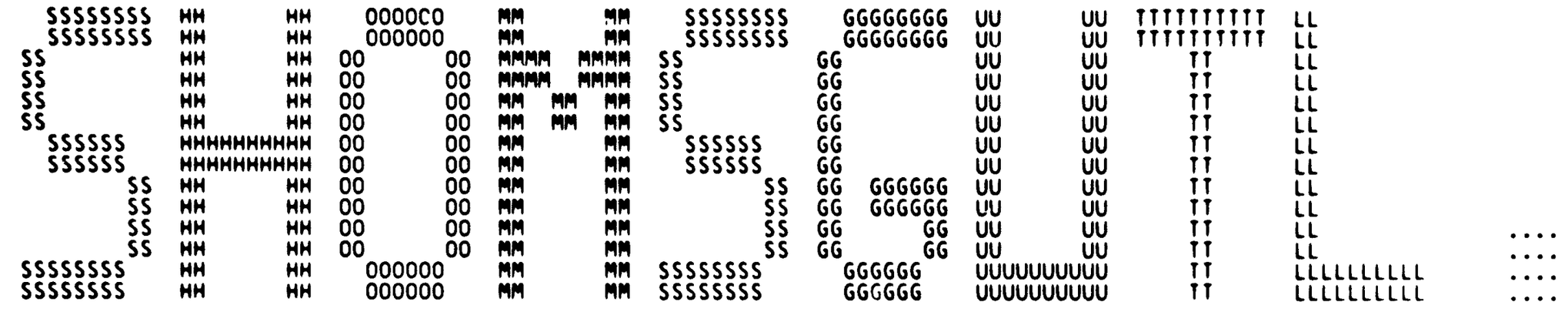

Table of contents

```
0000     1          .TITLE  SHOW$MSG_UTIL - MESSAGE PRINTING UTILITIES
0000     2          .IDENT  'V04-000'
0000     3
0000     4 ;
0000     5 ;****************************************************************************
0000     6 ;*                                                                          *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000     9 ;*  ALL RIGHTS RESERVED.                                                    *
0000    10 ;*                                                                          *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    16 ;*  TRANSFERRED.                                                            *
0000    17 ;*                                                                          *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000    20 ;*  CORPORATION.                                                            *
0000    21 ;*                                                                          *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    24 ;*                                                                          *
0000    25 ;*                                                                          *
0000    26 ;****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY:     SHOW COMMAND, MESSAGE UTILITY ROUTINES
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;
0000    34 ;       This module contains all message formatting and printing utilities
0000    35 ;       for the SHOW command
0000    36 ;
0000    37 ; ENVIRONMENT:
0000    38 ;
0000    39 ;       AUTHOR: BEATRICE U.WALTHER,     CREATION DATE:  10-OCT-1977
0000    40 ;
0000    41 ; MODIFIED BY:
0000    42 ;
0000    43 ;       V03-002 GAS0100         Gerry Smith             11-Jan-1983
0000    44 ;               Change SHOW$PRINT_LINE to SHOW_PRINT_LINE.
0000    45 ;
0000    46 ;       V03-001 GAS0065         Gerry Smith             29-Mar-1982
0000    47 ;               If an error occurs during open or write to SYS$OUTPUT,
0000    48 ;               signal the error and stop.
0000    49 ;
0000    50 ;       . : VERSION
0000    51 ; 01    -
0000    52 ;
0000    53 ; 02    Gary Fowler     22-May-1979
0000    54 ;       Remove FOP from $FAB so new output file will be created if
0000    55 ;       SYS$OUTPUT is reassigned to a file that already exists.
0000    56 ;
0000    57 ; 03    Jim Teague
```

```
0000     58 ;       Changed W^ references to L^ references.
0000     59 ;
0000     60 ;--
```

```
                        0000     62          .SBTTL  DECLARATIONS
                        0000     63
                        0000     64 ;
                        0000     65 ; INCLUDE FILES:
                        0000     66 ;
                        0000     67          $RMSDEF
                        0000     68 ;
                        0000     69 ; MACROS:
                        0000     70 ;
                        0000     71
                        0000     72
                        0000     73 ;
                        0000     74 ; EQUATED SYMBOLS:
                        0000     75 ;
                        0000     76 ;
                        0000     77 ; define structure of a message table entry
                        0000     78 ;
                        0000     79
                        0000     80 $DEFINI MSG
                        0000     81 $DEF     MSG_Q_TEXT:
00000008                0000     82          .BLKQ   1                   ; descriptor to message text
                        0008     83 $DEF     MSG_B_CODE:
00000009                0008     84          .BLKB   1                   ; code associated with message
                        0009     85 $DEF     MSG_B_CONTIN:
0000000A                0009     86          .BLKB   1                   ; continuation flag
                        000A     87 $DEF     MSG_B_PARAM:
0000000B                000A     88          .BLKB   1                   ; count of associated FAO parameters
0000000C                000B     89          .BLKB   1                   ; (spare)
                        000C     90 $DEF     MSG_K_LENGTH:
                        000C     91                                      ; length of message table entry
                        000C     92 $DEFEND MSG
                        0000     93
                        0000     94 ;
                        0000     95 ; define shared messages for open/write errors
                        0000     96 ;
                        0000     97 $SHR_MSGDEF      SHOW, 120, LOCAL,-      ; make it a SHOW error
                        0000     98                  <<OPENOUT,ERROR>,-      ; error opening file
                        0000     99                  <WRITEERR,ERROR>>       ; error writing to file
                        0000    100
                        0000    101 ;
                        0000    102 ; OWN STORAGE:
                        0000    103 ;
                        0000    104
                    00000000    105          .PSECT  SHOW$RWDATA     LONG,RD,WRT,NOEXE
                        0000    106
                        0000    107 OUTPUT_DNA:                                     ; FILE NAME STRING
3A 54 55 50 54 55 4F 24 53 59 53 0000    108          .ASCII  /SYS$OUTPUT:/
                    0000000B    0008    109 OUTPUT_DNA_LEN =.-OUTPUT_DNA
                        0008    110          .ALIGN  LONG
                        000C    111 OUTPUT_FAB:
                        000C    112          $FAB    FAC=PUT,-                  ; ALLOCATE FILE ACCESS BLOCK
                        000C    113                  DNA=OUTPUT_DNA,-           ; FOR OUTPUT
                        000C    114                  DNS=OUTPUT_DNA_LEN,-
                        000C    115                  NAM=OUTPUT_NAM,-           ; PROVIDE A NAME BLOCK
                        000C    116                  RAT=CR                     ; APPEND A CARRIAGE RETURN
                        005C    117
                        005C    118 OUTPUT_RAB:
```

```
                     005C   119          $RAB    FAB=OUTPUT_FAB                  ; AND RECORD ACCESS BLOCK
                     00A0   120 OUTPUT_NAM:
                     00A0   121          $NAM    ESA=OUT_ESA,-                   ; EXPANDED NAME
                     00A0   122                  ESS=NAM$C_MAXRSS,-
                     00A0   123                  RSA=OUT_RSA,-                   ; RESULTANT NAME
                     00A0   124                  RSS=NAM$C_MAXRSS
                     0100   125 OUT_RSA:
000001FF             0100   126          .BLKB   NAM$C_MAXRSS                    ; RESERVE SPACE FOR RESULTANT STRING
                     01FF   127 OUT_ESA:
000002FE             01FF   128          .BLKB   NAM$C_MAXRSS                    ; RESERVE SPACE FOR EXPANDED STRING
                     02FE   129 RMS_STATUS:
00000302             02FE   130          .BLKL   1                               ; RESERVE LONGWORD FOR RMS STATUS CO
                     0302   131 OUTPUT_LENGTH:
00000306             0302   132          .BLKL   1                               ; ACTUAL LENGTH OF OUTPUT BUFFER
                     0306   133 SHOW$GQ_OUT_DSC::
00000100'            0306   134          .LONG   OUT_BUFF_LENGTH                 ; BUILD DESCRIPTOR FOR OUTPUT  BUFFE
0000030E'            030A   135          .LONG   OUTPUT_BUFFER                   ;
                     030E   136 OUTPUT_BUFFER:
0000040E             030E   137          .BLKB   256                             ; ALLOCATE OUTPUT BUFFER
00000100             040E   138 OUT_BUFF_LENGTH =.-OUTPUT_BUFFER
```

```
                         040E          .SBTTL  SHOW$PRINT_MSG          - PRINT MESSAGE ROUTINE
                         040E
                         040E    ;++
                         040E    ; FUNCTIONAL DESCRIPTION:
                         040E    ;         THIS ROUTINE FORMATS ALL MESSAGES AND PRINTS THEM ON SYS$OUTPUT:
                         040E    ;         OR SYS$ERROR DEPENDING ON THE CODE ASSOCIATED WITH THE MESSAGE.
                         040E    ;         ( SEE DESCRIPTION OF MACRO SHOW_DEFMSG FOR FORMAT OF MESSAGE TABLE)
                         040E    ;
                         040E    ; CALLING SEQUENCE:
                         040E    ;
                         040E    ;                 PUSHL   #SHOW$_<MESSAGE MNEMONIC>
                         040E    ;                 PUSHAL  ARGLIST
                         040E    ;                 CALLS #2,SHOW$PRINT_MSG
                         040E    ;
                         040E    ; INPUT PARAMETERS:
                         040E    ;
                         040E    ;         4(AP)=  MESSAGE IDENTIFICATION NUMBER
                         040E    ;         8(AP)=  POINTER TO ARGUMENT LIST FOR $FAOL
                         040E    ;
                         040E    ; OUTPUT PARAMETERS:
                         040E    ;
                         040E    ;         NONE
                         040E    ;
                         040E    ; IMPLICIT INPUT PARAMETERS:
                         040E    ;
                         040E    ;         NONE
                         040E    ;
                         040E    ; IMPLICIT OUTPUT PARAMETERS:
                         040E    ;
                         040E    ;         CALLS ROUTINE SHOW_PRINT_LINE
                         040E    ;
                         040E    ; COMPLETION CODES:
                         040E    ;
                         040E    ;         NONE
                         040E    ;
                         040E    ; SIDE EFFECTS:
                         040E    ;
                         040E    ;         NONE
                         040E    ;--
                         040E
                     00000000          .PSECT  SHOW$CODE       BYTE,RD,NOWRT,EXE
                         0000
                   0C00  0000          .ENTRY  SHOW$PRINT_MSG,^M<R10,R11>
                         0002
                         0002    ;
                         0002    ; initialize
                         0002    ;
                         0002
                 01  BB  0002          PUSHR   #^M<R0>                 ; save status code
 5B    0C    08 AC   C5  0004          MULL3   8(AP),#MSG_K_LENGTH,R11
                         0009                                          ; CALCULATE OFFSET INTO MESSAGE TABLE
       5A    04 AC   D0  0009          MOVL    4(AP),R10               ; POINTER TO FAO PARAMETER LIST
                         000D
                         000D    ;
                         000D    ; format a message line
                         000D    ;
                         000D
```

```
                                000D   197
                                000D   198 10$:
                                000D   199          $FAOL_S         -                       ; FORMAT MESSAGE INTO OUPUT BUFFER
                                000D   200                  CTRSTR=L^SHOW$A_MSGTXT(R11), -  ; FAO CONTROL STRING
                                000D   201                  OUTLEN=L^OUTPUT_LENGTH, -               ; WORD TO RECEIVE ACTUAL LEN
                                000D   202                  OUTBUF=L^SHOW$G0_OUT_DSC, -     ; OUTPUT BUFFER DESCRIPTOR
                                000D   203                  PRMLST=(R10)                    ; PARAMETER LIST
                      28 50     E9  0028   204          BLBC    R0,99$                  ; branch on error
                                0028   205
                                0028   206 ;
                                0028   207 ; print formatted line
                                0028   208 ;
                                0028   209
                                0028   210
                    030E'CF     DF  002B   211          PUSHAL  W^OUTPUT_BUFFER         ; ADDRESS OF OUTPUT MESSAGE STRING
                    0302'CF     DD  002F   212          PUSHL   W^OUTPUT_LENGTH         ; RESULTANT LENGTH OF OUTPUT MESSAGE
                0054'CF    02   FB  0033   213          CALLS   #2,W^SHOW_PRINT_LINE    ; PRINT LINE ON SYS$OUTPUT:
                      18 50     E9  0038   214          BLBC    R0,99$                  ; branch on error
                                003B   215
                                003B   216 ;
                                003B   217 ; get next line of message
                                003B   218 ;
                                003B   219
                                003B   220
              0F 00000015'EB    E9  003B   221          BLBC    L^SHOW$A_MSGTXT+MSG_K_LENGTH+MSG_B_CONTIN(R11),90$
                                0042   222                                          ; NO CONTINUATION
           50   0000000A'EB    9A  0042   223          MOVZBL  L^SHOW$A_MSGTXT+MSG_B_PARAM(R11),R0
                                0049   224                                          ; SKIP FAO PARAMETERS ALREADY PROCESSED
                         5A 50  C0  0049   225          ADDL2   R0,R10                  ; UPDATE POINTER TO PARAMETER LIST
                         5B 0C  C0  004C   226          ADDL2   #MSG_K_LENGTH,R11       ; SKIP TO NEXT MESSAGE ENTRY
                            BC  11  004F   227          BRB     10$                     ; AND PROCESS NEXT LINE OF MESSAGE
                                0051   228
                                0051   229 ;
                                0051   230 ; exits
                                0051   231 ;
                                0051   232
                                0051   233 90$:
                01              BA  0051   234          POPR    #^M<R0>                 ; restore return code
                                04  0053   235 99$:     RET                             ; ALL DONE
```

```
                     0054   237          .SBTTL  SHOW_PRINT_LINE          - PRINT LINE ROUTINE
                     0054   238
                     0054   239 ;++
                     0054   240 ; FUNCTIONAL DESCRIPTION:
                     0054   241 ;
                     0054   242 ;       THIS ROUTINE PRINTS A LINE ON DEVICE SYS$OUTPUT
                     0054   243 ;
                     0054   244 ; CALLING SEQUENCE:
                     0054   245 ;
                     0054   246 ;               PUSHAL  OUT_BUFFER
                     0054   247 ;               PUSHL   OUT_LENGTH
                     0054   248 ;               CALLS   #2,SHOW_PRINT_LINE
                     0054   249 ;
                     0054   250 ;       WHERE:  OUT_LENGTH =  LENGTH OF CHARACTER STRING TO PRINT
                     0054   251 ;               OUT_BUFFER =  ADDRESS OF STRING TO PRINT
                     0054   252 ;
                     0054   253 ;       A CR/LF IS AUTOMATICALLY APPENDED FOR EACH CALL
                     0054   254 ;
                     0054   255 ;       NOTE:
                     0054   256 ;       IT IS HIGHLY RECOMMENDED TO USE THE COMMON LINE BUFFER PROVIDED
                     0054   257 ;        THE DESCRIPTOR OF WHICH IS ADDRESSED BY GLOBAL SYMBOL:
                     0054   258 ;                       SHOW$GQ_OUT_DSC
                     0054   259 ;
                     0054   260 ;
                     0054   261 ;
                     0054   262 ; INPUT PARAMETERS:
                     0054   263 ;
                     0054   264 ;               OUT_BUFFER
                     0054   265 ;               OUT_LENGTH
                     0054   266 ;
                     0054   267 ; IMPLICIT INPUTS:
                     0C54   268 ;
                     0054   269 ;       NONE
                     0054   270 ;
                     0054   271 ; OUTPUT PARAMETERS:
                     0054   272 ;
                     0054   273 ;       NONE
                     0054   274 ;
                     0054   275 ; IMPLICIT OUTPUTS:
                     0054   276 ;
                     0054   277 ;       LINE IS PRINTED ON SYS$OUTPUT:
                     0054   278 ;
                     0054   279 ; COMPLETION CODES:
                     0054   280 ;
                     0054   281 ;       RMS COMPLETION CODES
                     0054   282 ;
                     0054   283 ; SIDE EFFECTS:
                     0054   284 ;
                     0054   285 ;
                     0054   286 ;       OPENS AND CONNECTS SYS$OUTPUT: IF NECESSARY
                     0054   287 ;
                     0054   288 ;--
                     0054   289
                00000054    290          .PSECT  SHOW$CODE        BYTE,RD,NOWRT,EXE
                     0054   291
                000C 0054   292          .ENTRY  SHOW_PRINT_LINE,^M<R2,R3>
                     0056   293
```

```
                                0056   294 ;
                                0056   295 ; open SYS$OUTPUT: if not opened
                                0056   296 ;
                                0056   297
                 005E'CF    B5  0056   298          TSTW    W^RAB$W_ISI+OUTPUT_RAB  ; HAS OUTPUT ALREADY BEEN OPENED ?
                       23   12  005A   299          BNEQ    10$                     ; YES
      52 007810A2      8F   D0  005C   300          MOVL    #SHOW$_OPENOUT,R2       ; ASSUME ERROR OPENING
                                0063   301          $CREATE FAB=W^OUTPUT_FAB                ; NO, OPEN IT
                    30 50   E9  006E   302          BLBC    R0,100$                 ; BRANCH ON ERROR
                                0071   303          $CONNECT RAB=W^OUTPUT_RAB               ; AND CONNECT
                    22 50   E9  007C   304          BLBC    R0,100$                 ; branch on error
                                007F   305
                                007F   306 ;
                                007    307 ; print line on SYS$OUTPUT:
                                007F   308 ;
                                007F   309
                                007F   310 10$:
       0084'CF      08 AC   D0  007F   311          MOVL    8(AP),W^OUTPUT_RAB+RAB$L_RBF
                                0085   312                                          ; INSERT OUTPUT BUFFER ADDRESS
       007E'CF      04 AC   B0  0085   313          MOVW    4(AP),W^OUTPUT_RAB+RAB$W_RSZ
                                008B   314                                          ; INSERT OUTPUT BUFFER LENGTH
      52 007810D2      8F   D0  008B   315          MOVL    #SHOW$_WRITEERR,R2      ; ASSUME ERROR WRITING
                                0092   316          $PUT    RAB=W^OUTPUT_RAB        ; ISSUE RMS CALL TO PUT LINE
                    01 50   E9  009D   317          BLBC    R0,100$                 ; BRANCH IF ERROR
                                00A0   318 ;
                                00A0   319 ; exit
                                00A0   320 ;
                                00A0   321
                                00A0   322 90$:
                          04    00A0   323          RET                             ; RETURN
                                00A1   324
                                00A1   325 ;
                                00A1   326 ; if an error occurred doing I/O, signal it here and stop
                                00A1   327 ;
                                00A1   328
                                00A1   329 100$:
                       50   DD  00A1   330          PUSHL   R0                      ; PUSH ERROR STATUS
                                00A3   331 ;
                                00A3   332 ; check for non-zero RSL.  If no resultant name string, try the
                                00A3   333 ; expanded name string.  If both fail, simply use SYS$OUTPUT
                                00A3   334 ;
00000306'EF 000000A3'EF      9A 00A3   335          MOVZBL  OUTPUT_NAM+NAM$B_RSL,SHOW$GQ_OUT_DSC
                          0D  13  00AE   336          BEQL    110$
0000030A'EF 00000100'EF      DE 00B0   337          MOVAL   OUT_RSA,SHOW$GQ_OUT_DSC+4
                          2C  11  00BB   338          BRB     130$
                                00BD   339 110$:
00000306'EF 000000AB'EF      9A 00BD   340          MOVZBL  OUTPUT_NAM+NAM$B_ESL,SHOW$GQ_OUT_DSC
                          0D  13  00C8   341          BEQL    120$
0000030A'EF 000001FF'EF      DE 00CA   342          MOVAL   OUT_ESA,SHOW$GQ_OUT_DSC+4
                          12  11  00D5   343          BRB     130$
                                00D7   344 120$:
          00000306'EF     0B   D0 00D7   345          MOVL    #OUTPUT_DNA_LEN,SHOW$GQ_OUT_DSC
0000030A'EF 00000000'EF      DE 00DE   346          MOVAL   OUTPUT_DNA,SHOW$GQ_OUT_DSC+4
                                00E9   347 130$:
                00000306'EF  DF  00E9   348          PUSHAL  SHOW$GQ_OUT_DSC         ; PUSH OUTPUT DESCRIPTOR
                          01 DD  00EF   349          PUSHL   #1
                          52 DD  00F1   350          PUSHL   R2                      ; PUSH ERROR MESSAGE
```

```
                00000000'GF   04   FB  00F3   351          CALLS   #4,G^LIB$STOP          ; SIGNAL AND STOP
                                      00FA   352          .END
```

```
$$.TAB                          = 000000A0 R      02
$$.TABEND                       = 00000100 R      02
$$.TMP                          = 00000000
$$.TMP1                         = 00000001
$$.TMP2                         = 000000CF
$$GBL                           = 00000000
FAB$C_BID                       = 00000003
FAB$C_BLN                       = 00000050
FAB$C_SEQ                       = 00000000
FAB$C_VAR                       = 00000002
FAB$L_ALQ                       = 00000010
FAB$L_FOP                       = 00000004
FAB$V_CHAN_MODE                 = 00000002
FAB$V_CR                        = 00000001
FAB$V_FILE_MODE                 = 00000004
FAB$V_LNM_MODE                  = 00000000
FAB$V_PUT                       = 00000000
FAB$W_GBC                       = 00000048
LIB$STOP                          ********    X   03
MSG_B_CODE                        00000008  G
MSG_B_CONTIN                      00000009  G
MSG_B_PARAM                       0000000A  G
MSG_K_LENGTH                      0000000C  G
MSG_Q_TEXT                        0000000C  G
NAM$B_ESL                       = 0000000B
NAM$B_ESS                       = 0000000A
NAM$B_NOP                       = 00000008
NAM$B_RSL                       = 00000003
NAM$B_RSS                       = 00000002
NAM$C_BID                       = 00000002
NAM$C_BLN                       = 00000060
NAM$C_MAXRSS                    = 000000FF
NAM$L_ESA                       = 0000000C
NAM$L_RSA                       = 00000004
OUTPUT_BUFFER                     0000030E R      02
OUTPUT_DNA                        00000000 R      02
OUTPUT_DNA_LEN                  = 0000000B
OUTPUT_FAB                        0000000C R      02
OUTPUT_LENGTH                     00000302 R      02
OUTPUT_NAM                        000000A0 R      02
OUTPUT_RAB                        0000005C R      02
OUT_BUFF_LENGTH                 = 00000100
OUT_ESA                           000001FF R      02
OUT_RSA                           00000100 R      02
RAB$B_RAC                       = 0000001E
RAB$C_BID                       = 00000001
RAB$C_BLN                       = 00000044
RAB$C_SEQ                       = 00000000
RAB$L_CTX                       = 00000018
RAB$L_RBF                       = 00000028
RAB$L_ROP                       = 00000004
RAB$W_ISI                       = 00000002
RAB$W_RSZ                       = 00000022
RMS_STATUS                        000002FE R      02
SHOW$A_MSGTXT                     ********    X   03
SHOW$GQ_OUT_DSC                   00000306 RG     02
SHOW$PRINT_MSG                    00000000 RG     03
SHOW$_OPENOUT                   = 007810A2
SHOW$_WRITEERR                  = 007810D2
SHOW_PRINT_LINE                   00000054 RG     03
SHR$K_SHRDEF                    = 00000001
SHR$_OPENOUT                    = 000010A0
SHR$_WRITEERR                   = 000010D0
SYS$CONNECT                       ********  GX    03
SYS$CREATE                        ********  GX    03
SYS$FAOL                          ********  GX    03
SYS$PUT                           ********  GX    03
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	0000000C (12.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SHOW$RWDATA	0000040E (1038.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SHOW$CODE	000000FA (250.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.06	00:00:01.10
Command processing	83	00:00:00.89	00:00:07.17
Pass 1	262	00:00:08.71	00:00:32.97
Symbol table sort	0	00:00:00.94	00:00:03.20
Pass 2	72	00:00:01.62	00:00:05.92
Symbol table output	8	00:00:00.09	00:00:00.49
Psect synopsis output	3	00:00:00.03	00:00:00.09
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	440	00:00:12.35	00:00:50.95

The working set limit was 1200 pages.
44030 bytes (86 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 820 non-local and 9 local symbols.
352 source lines were read in Pass 1, producing 21 object records in Pass 2.
32 pages of virtual memory were used to define 25 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[CLIUTL.OBJ]CLIUTL.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	21
TOTALS (all libraries)	21

1151 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SHOMSGUTL/OBJ=OBJ$:SHOMSGUTL MSRC$:SHOMSGUTL/UPDATE=(ENH$:SHOMSGUTL)+EXECML$/LIB+LIB$:CLIUTL/LIB
```

SHOMSGUTL LIS

SHONET LIS

SHOWAUDIT LIS

SHOWID LIS

SHOWLOG LIS

SHOWERROR LIS

SHOWFILES LIS

SHOMEMORY LIS