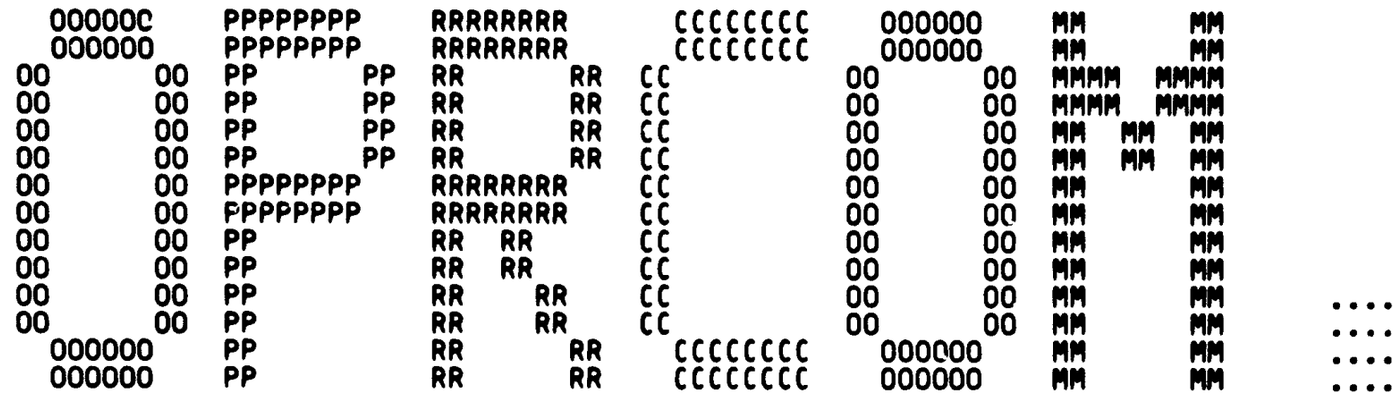
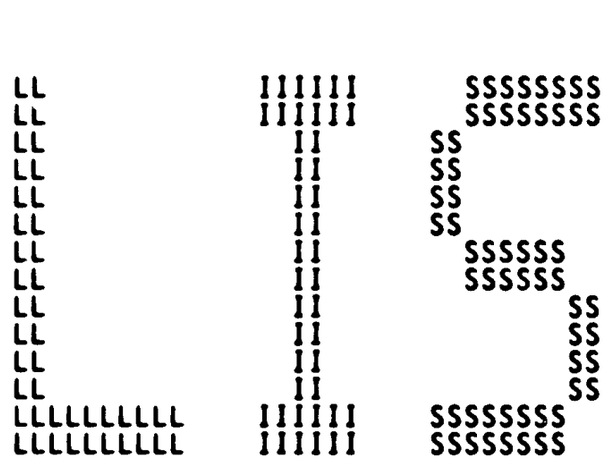

MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP

FILEID**OPRCOM





```
;  1          0001  0
;  2          0002  0 MODULE OPRCOM (LANGUAGE (BLISS32) ,
;  3          0003  0                IDENT = 'V04-000'
;  4          0004  0                ) =
;  5          0005  1 BEGIN
;  6          0006  1 !
;  7          0007  1 !*****************************************************************************
;  8          0008  1 !*                                                                           *
;  9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
; 10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
; 11          0011  1 !*  ALL RIGHTS RESERVED.                                                    *
; 12          0012  1 !*                                                                         *
; 13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 18          0018  1 !*  TRANSFERRED.                                                            *
; 19          0019  1 !*                                                                         *
; 20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 22          0022  1 !*  CORPORATION.                                                            *
; 23          0023  1 !*                                                                         *
; 24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
; 26          0026  1 !*                                                                         *
; 27          0027  1 !*                                                                         *
; 28          0028  1 !*****************************************************************************
; 29          0029  1
; 30          0030  1 !++
; 31          0031  1 !
; 32          0032  1 ! FACILITY:  MTAACP
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !       This module communicates with the operator
; 36          0036  1 !
; 37          0037  1 !
; 38          0038  1 ! ENVIRONMENT:
; 39          0039  1 !
; 40          0040  1 !       VMS operating system, including privileged system services
; 41          0041  1 !       and internal exec routines.
; 42          0042  1 !
; 43          0043  1 !--
; 44          0044  1
; 45          0045  1 !
; 46          0046  1 !
; 47          0047  1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  24-AUG-1977
; 48          0048  1 !
; 49          0049  1 ! MODIFIED BY:
; 50          0050  1 !
; 51          0051  1 !       V03-002 HH0041          Hai Huang                   24-Jul-1984
; 52          0052  1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
; 53          0053  1 !
; 54          0054  1 !       V03-001 MMD0102         Meg Dumont,     17-Feb-1983  13:12
; 55          0055  1 !               Use GET_DEV_NAME to get the tape units' device name. Add
; 56          0056  1 !               check for OPCOM, so that the MTAACP can recover if OPCOM
; 57          0057  1 !               is not available.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-006 DMW00078        David Michael Walp      8-Feb-1982
;    60          0060  1 !               Fixed pushing and poping of Username and Account CTL
;    61          0061  1 !               fields around opreator request.
;    62          0062  1 !
;    63          0063  1 !       V03-005 DMW00021        David Michael Walp      26-May-1981
;    64          0064  1 !               Move GOLBAL LITERAL WORK_AREA_SZ to MTADEF.
;    65          0065  1 !
;    66          0066  1 !       V02-004 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    67          0067  1 !
;    68          0068  1 !**
;    69          0069  1
;    70          0070  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    71          0071  1
;    72          0072  1 REQUIRE 'SRC$:MTADEF.B32';
;    73          0456  1
;    74          0457  1
;    75          0458  1 FORWARD ROUTINE
;    76          0459  1     CANCEL_OP_REPLY     : COMMON_CALL NOVALUE,  ! cancel reply from operator
;    77          0460  1     FILL_PROC   : COMMON_CALL NOVALUE,
;    78          0461  1     PRINT_NOT_LABEL     : NOVALUE JSB,          ! print not correct label
;    79          0462  1     PRINT_OPR_MSG       : L$PRINT_OPR_MSG,      ! print operator message
;    80          0463  1     SEND_MSG    : COMMON_CALL;                  ! send msg to tape operator
;    81          0464  1
;    82          0465  1 GLOBAL
;    83          0466  1     WORK_AREA   : VECTOR [WORK_AREA_SZ, BYTE];
;    84          0467  1
;    85          0468  1 EXTERNAL
;    86          0469  1     CTL$T_USERNAME      : ADDRESSING_MODE (ABSOLUTE),
;    87          0470  1     CTL$T_ACCOUNT       : ADDRESSING_MODE (ABSOLUTE),
;    88          0471  1     CURRENT_UCB : REF BBLOCK,                   ! address of current unit control block
;    89          0472  1     MAIL_CHANNEL;                               ! channel for reply mailbox
;    90          0473  1
;    91          0474  1  EXTERNAL ROUTINE
;    92          0475  1     GET_DEV_NAME        : COMMON_CALL NOVALUE;  ! Given the UCB add get dev name
;    93          0476  1
;    94          0477  1
;    95          0478  1 ! SETUP MESSAGE DESCRIPTOR AND BUFFER IN WORK AREA
;    96          0479  1
;    97          0480  1 BIND
;    98          0481  1     DESCR = WORK_AREA   : VECTOR [2],
;    99          0482  1     MSG = WORK_AREA + 8 : BBLOCK;
;   100          0483  1
```

```
;  102     0484  1 ROUTINE SEND_MSG (MB_CHANNEL) : COMMON_CALL =
;  103     0485  1
;  104     0486  1 !++
;  105     0487  1 !
;  106     0488  1 ! FUNCTIONAL DESCRIPTION:
;  107     0489  1 !       This routine formats and sends a message to the tape operator
;  108     0490  1 !
;  109     0491  1 ! CALLING SEQUENCE:
;  110     0492  1 !       SEND_MSG()
;  111     0493  1 !
;  112     0494  1 ! INPUT PARAMETERS:
;  113     0495  1 !       ARG1 - mailbox channel number
;  114     0496  1 !
;  115     0497  1 ! IMPLICIT INPUTS:
;  116     0498  1 !       CURRENT_VCB - address of current volume control block
;  117     0499  1 !       DESCR[0] - length of text to be sent
;  118     0500  1 !       MSG[OPC$L_MS_TEXT] - text to be sent
;  119     0501  1 !
;  120     0502  1 ! OUTPUT PARAMETERS:
;  121     0503  1 !       NONE
;  122     0504  1 !
;  123     0505  1 ! IMPLICIT OUTPUTS:
;  124     0506  1 !       message sent
;  125     0507  1 !
;  126     0508  1 ! ROUTINE VALUE:
;  127     0509  1 !       SS$_DEVOFFLIN - no operator
;  128     0510  1 !       SS$_MBTOOSML  - mailbox too small for message
;  129     0511  1 !
;  130     0512  1 ! SIDE EFFECTS:
;  131     0513  1 !       NONE
;  132     0514  1 !
;  133     0515  1 !--
;  134     0516  1
;  135     0517  2     BEGIN
;  136     0518  2
;  137     0519  2     EXTERNAL REGISTER
;  138     0520  2         COMMON_REG;
;  139     0521  2
;  140     0522  2     EXTERNAL
;  141     0523  2         ACP_USERNAME : VECTOR [ , BYTE ],
;  142     0524  2         ACP_ACCOUNT  : VECTOR [ , BYTE ];
;  143     0525  2
;  144     0526  2     BIND
;  145     0527  2         SECONDS = UPLIT (-10000000, -1);
;  146     0528  2
;  147     0529  2     LOCAL
;  148     0530  2         VVP    : REF BBLOCK,
;  149     0531  2         STATUS ;                                ! status of mailbox write
;  150     0532  2
;  151     0533  2     ! format message header
;  152     0534  2     !
;  153     0535  2     MSG[OPC$B_MS_TARGET] = OPR$M_TAPES;
;  154     0536  2     MSG[OPC$L_MS_RPLYID] = .CURRENT_VCB;        ! id if reply
;  155     0537  2     DESCR[0] = .DESCR[0] + $BYTEOFFSET(OPC$L_MS_TEXT);
;  156     0538  2     DESCR[1] = MSG;
;  157     0539  2     VVP = .CURRENT_VCB [ VCB$L_VPFL ];
;  158     0540  2
```

```
;  159          0541  2     INCR J FROM 0 TO 119 DO                    ! wait two minutes
;  160          0542  2         BEGIN
;  161          0543  2
;  162          0544  2         ! make it look like the user is making the opcom request
;  163          0545  2         !
;  164          0546  2         KERNEL_CALL ( FILL_PROC, VVP[VVP$T_USERNAME], VVP[VVP$T_ACCOUNT] );
;  165          0547  2
;  166          0548  2         STATUS = $SNDOPR ( MSGBUF = DESCR, CHAN = .MB_CHANNEL );
;  167          0549  2
;  168          0550  2         KERNEL_CALL ( FILL_PROC, ACP_USERNAME, ACP_ACCOUNT );
;  169          0551  2
;  170          0552  2         ! If the status is opc$_noperator then reset the error to be
;  171          0553  2         ! the fatal error. This means that there is no opcom available
;  172          0554  2         ! as would be the case if we are running on Midnite.
;  173          0555  2
;  174          0556  3         IF .STATUS EQL OPC$_NOPERATOR THEN
;  175          0557  4            BEGIN
;  176          0558  4                 BIND
;  177          0559  4                 STATUS2 = STATUS : BLOCK [4, BYTE];
;  178          0560  4                 STATUS2 [STS$V_SEVERITY] = STS$K_ERROR;
;  179          0561  4                 EXITLOOP;
;  180          0562  3             END;
;  181          0563  3
;  182          0564  3         ! Return to the caller if the send to operator worked. Or if it
;  183          0565  3         ! got an unrecoverable error. Device off line means there is
;  184          0566  3         ! no opcom available as would be the case if we are running on
;  185          0567  3         ! Midnite.
;  186          0568  3
;  187          0569  3         IF .STATUS
;  188          0570  3             OR
;  189          0571  4             (.STATUS NEQ SS$_INSFMEM
;  190          0572  4             AND
;  191          0573  4             .STATUS NEQ SS$_MBFULL)
;  192          0574  4            OR (.STATUS EQL SS$_DEVOFFLINE)
;  193          0575  3         THEN
;  194          0576  3             EXITLOOP;                          ! if temporary problem, then repeat
;  195          0577  3
;  196          0578  4         IF $SETIMR(DAYTIM = SECONDS, EFN = TIMEFN)
;  197          0579  3         THEN
;  198          0580  3             $WAITFR(EFN = TIMEFN);
;  199          0581  3
;  200          0582  2         END;
;  201          0583  2
;  202          0584  2     RETURN .STATUS;
;  203          0585  2
;  204          0586  1     END;                                       ! end of routine

                                                        .TITLE  OPRCOM
                                                        .IDENT  \V04-000\
                                                        .PSECT  $CODE$,NOWRT,2
                         FFFFFFFF  FF676980  00000 P.AAA:  .LONG   -10000000, -1                    ;
                                                        .PSECT  $LOCKEDD1$,NOEXE,2
```

```
                        00000 WORK_AREA::
                                .BLKB   128

                              DESCR=            WORK_AREA
                              MSG=              WORK_AREA+8
                              SECONDS=          P.AAA
                                .EXTRN  CTL$T_USERNAME, CTL$T_ACCOUNT
                                .EXTRN  CURRENT_UCB, MAIL_CHANNEL
                                .EXTRN  GET_DEV_NAME, ACP_USERNAME
                                .EXTRN  ACP_ACCOUNT, SYS$CMKRNL
                                .EXTRN  SYS$SNDOPR, SYS$SETIMR
                                .EXTRN  SYS$WAITFR

                                .PSECT  $CODE$,NOWRT,2

                        00FC 00000 SEND_MSG:
                                .WORD   Save R2,R3,R4,R5,R6,R7                  ; 0484
     57 00000000G   9F   9E 00002       MOVAB   @#SYS$CMKRNL, R7                ;
     56     0000'   CF   9E 00009       MOVAB   DESCR, R6                       ;
09   A6             04   90 0000E       MOVB    #4, MSG+1                       ; 0535
0C   A6             5B   D0 00012       MOVL    CURRENT_UCB, MSG+4              ; 0536
     66             08   C0 00016       ADDL2   #8, DESCR                       ; 0537
04   A6       08    A6   9E 00019       MOVAB   MSG, DESCR+4                    ; 0538
     50       3C    AB   D0 0001E       MOVL    60(CURRENT_UCB), VVP            ; 0539
     55     01BC    C0   9E 00022       MOVAB   444(R0), R5                     ; 0546
     54     01B0    C0   9E 00027       MOVAB   432(R0), R4                     ;
                    53   D4 0002C       CLRL    J                               ; 0548
                    30   BB 0002E 1$.    PUSHR   #^M<R4,R5>                      ; 0546
                    02   DD 00030       PUSHL   #2                              ;
                    5E   DD 00032       PUSHL   SP                              ;
              0000V CF   9F 00034       PUSHAB  FILL_PROC                       ;
          67        05   FB 00038       CALLS   #5, SYS$CMKRNL                  ;
            04      AC   DD 0003B       PUSHL   MB_CHANNEL                      ; 0548
                    56   DD 0003E       PUSHL   R6                              ;
 00000000G  00           02   FB 00040       CALLS   #2, SYS$SNDOPR              ;
          52            50   D0 00047       MOVL    R0, STATUS                  ;
              0000G CF   9F 0004A       PUSHAB  ACP_ACCOUNT                     ; 0550
              0000G CF   9F 0004E       PUSHAB  ACP_USERNAME                    ;
                    02   DD 00052       PUSHL   #2                              ;
                    5E   DD 00054       PUSHL   SP                              ;
              0000V CF   9F 00056       PUSHAB  FILL_PROC                       ;
          67        05   FB 0005A       CALLS   #5, SYS$CMKRNL                  ;
 00058061   8F           52   D1 0005D       CMPL    STATUS, #360545             ; 0556
                    07   12 00064       BNEQ    2$                              ;
52       03         00   02   F0 00066       INSV    #2, #0, #3, STATUS2         ; 0560
                    41   11 0006B       BRB     5$                              ; 0557
          3E            52   E8 0006D 2$:    BLBS    STATUS  5$                  ; 0569
 00000124   8F           52   D1 00070       CMPL    STATUS, #292                ; 0571
                    09   13 00077       BEQL    3$                              ;
 000008D8   8F           52   D1 00079       CMPL    STATUS, #2264               ; 0573
                    2C   12 00080       BNEQ    5$                              ;
 00000084   8F           52   D1 00082 3$:    CMPL    STATUS, #132                ; 0574
                    23   13 00089       BEQL    5$                              ;
                    7E   7C 0008B       CLRQ    -(SP)                           ; 0578
              FF67  CF   9F 0008D       PUSHAB  SECONDS                         ;
                    03   DD 00091       PUSHL   #3                              ;
 00000000G  00           04   FB 00093       CALLS   #4, SYS$SETIMR              ;
```

```
                                09              50  E9 0009A          BLBC    R0, 4$                          ;
                                                03  DD 0009D          PUSHL   #3                              ; 0580
                    00000000G   00              01  FB 0009F          CALLS   #1, SYS$WAITFR                  ;
                80              53 00000077     8F  F3 000A6 4$:      AOBLEQ  #119, J, 1$                     ; 0541
                                50              52  D0 000AE 5$:      MOVL    STATUS, R0                      ; 0584
                                                    04 000B1          RET                                     ; 0586

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 0008

;  205          0587  1
```

```
;  207     0588  1 GLOBAL ROUTINE PRINT_NOT_LABEL (MVL_ENTRY) : JSB NOVALUE =
;  208     0589  1
;  209     0590  1 !++
;  210     0591  1 !
;  211     0592  1 ! FUNCTIONAL DESCRIPTION:
;  212     0593  1 !       This routine prints at the tape operator's terminal the message
;  213     0594  1 !       'vol on mtax: not xxxx'
;  214     0595  1 !
;  215     0596  1 ! CALLING SEQUENCE:
;  216     0597  1 !       PRINT_NOT_LABEL(ARG1)
;  217     0598  1 !
;  218     0599  1 ! INPUT PARAMETERS:
;  219     0600  1 !       NONE
;  220     0601  1 !
;  221     0602  1 ! IMPLICIT INPUTS:
;  222     0603  1 !       CURRENT_UCB - address of current unit control block
;  223     0604  1 !       CURRENT_VCB - address of current volume control block
;  224     0605  1 !
;  225     0606  1 ! OUTPUT PARAMETERS:
;  226     0607  1 !       NONE
;  227     0608  1 !
;  228     0609  1 ! IMPLICIT OUTPUTS:
;  229     0610  1 !       Message printed at tape operator's terminal unless
;  230     0611  1 !       there is no operator or the mailbox is too small
;  231     0612  1 !
;  232     0613  1 ! ROUTINE VALUE:
;  233     0614  1 !       NONE
;  234     0615  1 !
;  235     0616  1 ! SIDE EFFECTS:
;  236     0617  1 !       NONE
;  237     0618  1 !
;  238     0619  1 !--
;  239     0620  1
;  240     0621  2     BEGIN
;  241     0622  2
;  242     0623  2     EXTERNAL REGISTER
;  243     0624  2         COMMON_REG;
;  244     0625  2
;  245     0626  2     LOCAL
;  246     0627  2         CVT_DEVNAM      : VECTOR [MAX_DEVNAM_LENGTH,BYTE], ! Converted dev name
;  247     0628  2         CVT_DEVNAM_LENGTH : BYTE,                  ! and length of dev name
;  248     0629  2         LABEL_ADDR      : REF VECTOR [, BYTE],  ! address of label
;  249     0630  2         LABEL_SZ;
;  250     0631  2
;  251     0632  2     MAP
;  252     0633  2         MVL_ENTRY       : REF BBLOCK;           ! address of volume entry in mvl
;  253     0634  2
;  254     0635  2     LABEL_ADDR = MVL_ENTRY[MVL$T_VOLLBL];
;  255     0636  2     LABEL_SZ = 0;
;  256     0637  2
;  257     0638  2     ! flush out trailing blanks
;  258     0639  2     !
;  259     0640  2
;  260     0641  2     INCR I FROM 0 TO MVL$S_VOLLBL - 1 DO
;  261     0642  3         BEGIN
;  262     0643  3
;  263     0644  3         IF .LABEL_ADDR[.I] EQL ' '
```

```
;  264          0645  3          THEN
;  265          0646  3              EXITLOOP;
;  266          0647  3
;  267          0648  3          LABEL_SZ = .LABEL_SZ + 1;
;  268          0649  2          END;
;  269          0650  2
;  270          0651  2      ! This next call will use the UCB address to get the device's name and
;  271          0652  2      ! will fill in the fields with that name and the length of the name.
;  272          0653  2
;  273          0654  2      GET_DEV_NAME(CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;  274          0655  2      PRINT_OPR_MSG(MOON$_NOTLABEL, 0, .LABEL_SZ, .LABEL_ADDR,
;  275          0656  2          .CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;  276          0657  1      END;                                    ! end of routine



                                    0C  BB 00000 PRINT_NOT_LABEL::
                                                     PUSHR   #^M<R2,R3>                      ; 0588
                              5E        14  C2 00002         SUBL2   #20, SP                 ;
                              52    20  AE  D0 00005         MOVL    MVL_ENTRY, LABEL_ADDR   ; 0635
                                        53  D4 00009         CLRL    LABEL_SZ                ; 0636
                                        50  D4 0000B         CLRL    I                       ; 0644
                              20      6042  91 0000D 1$:     CMPB    (I)[LABEL_ADDR], #32    ;
                                        06  13 00011         BEQL    2$                      ;
                                        53  D6 00013         INCL    LABEL_SZ                ; 0648
                          F4            50  F3 00015         AOBLEQ  #5, I, 1$               ; 0641
                                    04  AE  9F 00019 2$:     PUSHAB  CVT_DEVNAM              ; 0654
                                    04  AE  9F 0001C         PUSHAB  CVT_DEVNAM_LENGTH       ;
                                0000G   CF  02  FB 0001F     CALLS   #2, GET_DEV_NAME        ;
                                    04  AE  9F 00024         PUSHAB  CVT_DEVNAM              ; 0655
                              7E    04  AE  9A 00027         MOVZBL  CVT_DEVNAM_LENGTH, -(SP) ; 0656
                                        52  DD 0002B         PUSHL   LABEL_ADDR              ; 0655
                                        53  DD 0002D         PUSHL   LABEL_SZ                ;
                                        7E  D4 0002F         CLRL    -(SP)                   ;
                                00728104    8F  DD 00031     PUSHL   #7504132                ;
                                        0000V 30  00037      BSBW    PRINT_OPR_MSG           ;
                              5E            2C  C0 0003A     ADDL2   #44, SP                 ; 0657
                                            0C  BA 0003D     POPR    #^M<R2,R3>              ;
                                                05 0003F     RSB                             ;

; Routine Size:  64 bytes,    Routine Base:  $CODE$ + 00BA

;  277          0658  1
```

```
;  279          0659  1 GLOBAL ROUTINE CANCEL_OP_REPLY : COMMON_CALL NOVALUE =
;  280          0660  1
;  281          0661  1 !++
;  282          0662  1 !
;  283          0663  1 ! FUNCTIONAL DESCRIPTION:
;  284          0664  1 !       This routine cancels a request sent to the operator
;  285          0665  1 !
;  286          0666  1 ! CALLING SEQUENCE:
;  287          0667  1 !       CANCEL_OP_REPLY()
;  288          0668  1 !
;  289          0669  1 ! INPUT PARAMETERS:
;  290          0670  1 !       NONE
;  291          0671  1 !
;  292          0672  1 ! IMPLICIT INPUTS:
;  293          0673  1 !       CURRENT_VCB - address of current volume control block
;  294          0674  1 !
;  295          0675  1 ! OUTPUT PARAMETERS:
;  296          0676  1 !       NONE
;  297          0677  1 !
;  298          0678  1 ! IMPLICIT OUTPUTS:
;  299          0679  1 !       NONE
;  300          0680  1 !
;  301          0681  1 ! ROUTINE VALUE:
;  302          0682  1 !       NONE
;  303          0683  1 !
;  304          0684  1 ! SIDE EFFECTS:
;  305          0685  1 !       NONE
;  306          0686  1 !
;  307          0687  1 ! USER ERRORS:
;  308          0688  1 !       NONE
;  309          0689  1 !
;  310          0690  1 !--
;  311          0691  1
;  312          0692  2     BEGIN
;  313          0693  2
;  314          0694  2     EXTERNAL REGISTER
;  315          0695  2         COMMON_REG;
;  316          0696  2
;  317          0697  2     MSG[OPC$B_MS_TYPE] = OPC$_RQ_CANCEL;
;  318          0698  2     DESCR[0] = 0;                           ! no message text
;  319          0699  2     RETURN SEND_MSG(.MAIL_CHANNEL);
;  320          0700  2
;  321          0701  1     END;


                                        0000 00000          .ENTRY  CANCEL_OP_REPLY, Save nothing                   ; 0659
                    0000'  CF                   05  90 00002  MOVB    #5, MSG                                         ; 0697
                                        0000'   CF  D4 00007  CLRL    DESCR                                           ; 0698
                                        0000G   CF  DD 0000B  PUSHL   MAIL_CHANNEL                                    ; 0699
                    FEFA   CF                   01  FB 000CF  CALLS   #1, SEND_MSG
                                                    04 00014  RET                                                     ; 0701

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 00FA
```

```
;  322          0702  1
```

```
;  324     0703  1 GLOBAL ROUTINE PRINT_OPR_MSG (MSGNO, CHANNEL, ARGLIST) : L$PRINT_OPR_MSG =
;  325     0704  1
;  326     0705  1 !++
;  327     0706  1 !
;  328     0707  1 ! FUNCTIONAL DESCRIPTION:
;  329     0708  1 !       This routine gets a system message, formats the messages and
;  330     0709  1 !       prints it.
;  331     0710  1 !
;  332     0711  1 ! CALLING SEQUENCE:
;  333     0712  1 !       PRINT_OPR_MSG(ARG1,ARG2,ARG3)
;  334     0713  1 !
;  335     0714  1 ! INPUT PARAMETERS:
;  336     0715  1 !       ARG1 - system message number
;  337     0716  1 !       ARG2 - channel number for reply if one is wanted
;  338     0717  1 !       ARG3 - arguments for inclusion in message
;  339     0718  1 !
;  340     0719  1 ! IMPLICIT INPUTS:
;  341     0720  1 !       NONE
;  342     0721  1 !
;  343     0722  1 ! OUTPUT PARAMETERS:
;  344     0723  1 !       NONE
;  345     0724  1 !
;  346     0725  1 ! IMPLICIT OUTPUTS:
;  347     0726  1 !       NONE
;  348     0727  1 !
;  349     0728  1 ! ROUTINE VALUE:
;  350     0729  1 !       status code from send to operator
;  351     0730  1 !
;  352     0731  1 ! SIDE EFFECTS:
;  353     0732  1 !       DATA is used as a work area(also used by RDATTR)
;  354     0733  1 !
;  355     0734  1 !--
;  356     0735  1
;  357     0736  2     BEGIN
;  358     0737  2
;  359     0738  2     EXTERNAL REGISTER
;  360     0739  2         COMMON_REG;
;  361     0740  2
;  362     0741  2     EXTERNAL ROUTINE
;  363     0742  2         SYS$FAOL        : ADDRESSING_MODE (ABSOLUTE),
;  364     0743  2         SYS$GETMSG      : ADDRESSING_MODE (ABSOLUTE);
;  365     0744  2
;  366     0745  2     EXTERNAL
;  367     0746  2         DATA;
;  368     0747  2
;  369     0748  2     LOCAL
;  370     0749  2         DESCR_FAOSTR    : VECTOR [2];
;  371     0750  2
;  372     0751  2     DESCR_FAOSTR[0] = 256;
;  373     0752  2     DESCR_FAOSTR[1] = DATA;
;  374     0753  2     SYS$GETMSG(.MSGNO, DESCR_FAOSTR, DESCR_FAOSTR, 1, 0);
;  375     0754  2     DESCR[0] = WORK_AREA_SZ - 8 - $BYTEOFFSET(OPC$L_MS_TEXT);
;  376     0755  2     DESCR[1] = MSG[OPC$L_MS_TEXT];
;  377     0756  2     SYS$FAOL(DESCR_FAOSTR, DESCR, DESCR, ARGLIST);
;  378     0757  2     MSG[OPC$B_MS_TYPE] = OPC$_RQ_RQST;
;  379     0758  3     (
;  380     0759  3
```

```
;  381          0760  3     LOCAL
;  382          0761  3         STATUS;
;  383          0762  3
;  384          0763  3     STATUS = SEND_MSG(.CHANNEL);
;  385          0764  3
;  386          0765  3     IF NOT .STATUS
;  387          0766  3     THEN
;  388          0767  2         RETURN .STATUS);
;  389          0768  2
;  390          0769  2     RETURN 1;
;  391          0770  2
;  392          0771  1     END;                                    ! end of routine PRINT_OPR_MSG


                                                                    .EXTRN  SYS$FAOL, SYS$GETMSG
                                                                    .EXTRN  DATA

                         5E              04  C2 00000 PRINT_OPR_MSG::
                                                         SUBL2   #4, SP                          ; 0703
                         7E      0100    8F  3C 00003            MOVZWL  #256, DESCR_FAOSTR      ; 0751
                      04 AE      0000G   CF  9E 00008            MOVAB   DATA, DESCR_FAOSTR+4    ; 0752
                         7E              01  7D 0000E            MOVQ    #1, -(SP)               ; 0753
                           08    AE      9F  00011               PUSHAB  DESCR_FAOSTR            ;
                           0C    AE      9F  00014               PUSHAB  DESCR_FAOSTR            ;
                           1C    AE      DD  00017               PUSHL   MSGNO                   ;
                 00000000G   9F          05  FB 0001A            CALLS   #5, @#SYS$GETMSG        ;
                     0000'   CF    70    8F  9A 00021            MOVZBL  #112, DESCR             ; 0754
                     0000'   CF  0000'   CF  9E 00027            MOVAB   MSG+8, DESCR+4          ; 0755
                           14    AE      9F  0002E               PUSHAB  ARGLIST                 ; 0756
                       0000'   CF      9F  00031               PUSHAB  DESCR                   ;
                       0000'   CF      9F  00035               PUSHAB  DESCR                   ;
                           0C    AE      9F  00039               PUSHAB  DESCR_FAOSTR            ;
                 00000000G   9F          04  FB 0003C            CALLS   #4, @#SYS$FAOL          ;
                     0000'   CF          03  90 00043            MOVB    #3, MSG                 ; 0757
                           10    AE      DD  00048               PUSHL   CHANNEL                 ; 0763
                       FEA9    CF          01  FB 0004B            CALLS   #1, SEND_MSG            ;
                               03          50  E9 00050            BLBC    STATUS, T$              ; 0765
                               50          01  D0 00053            MOVL    #1, R0                  ; 0769
                               5E          08  C0 00056 1$:        ADDL2   #8, SP                  ; 0771
                                           05  00059               RSB                             ;

; Routine Size:  90 bytes,    Routine Base:  $CODE$ + 010F

;  393          0772  1
```

```
;  395          0773  1 ROUTINE FILL_PROC ( USERNAME, ACCOUNT ) : COMMON_CALL NOVALUE =
;  396          0774  1
;  397          0775  1 !++
;  398          0776  1 !
;  399          0777  1 ! FUNCTIONAL DESCRIPTION:
;  400          0778  1 !       This routine fills in the username and account in P1 space.  So
;  401          0779  1 !       Opcom messages look like they are coming from the user rather then
;  402          0780  1 !       ACP
;  403          0781  1 !
;  404          0782  1 ! CALLING SEQUENCE:
;  405          0783  1 !       FILE_PROC ( AGR1, ARG2 ) in kernel mode
;  406          0784  1 !
;  407          0785  1 ! INPUT PARAMETERS:
;  408          0786  1 !       ARG1 - address of username string
;  409          0787  1 !       ARG2 - address of account name string
;  410          0788  1 !
;  411          0789  1 ! IMPLICIT INPUTS:
;  412          0790  1 !       NONE
;  413          0791  1 !
;  414          0792  1 ! OUTPUT PARAMETERS:
;  415          0793  1 !       NONE
;  416          0794  1 !
;  417          0795  1 ! IMPLICIT OUTPUTS:
;  418          0796  1 !       NONE
;  419          0797  1 !
;  420          0798  1 ! ROUTINE VALUE:
;  421          0799  1 !       NONE
;  422          0800  1 !
;  423          0801  1 ! SIDE EFFECTS:
;  424          0802  1 !       NONE
;  425          0803  1 !
;  426          0804  1 !--
;  427          0805  1
;  428          0806  2     BEGIN
;  429          0807  2
;  430          0808  2     MAP
;  431          0809  2         USERNAME : REF VECTOR [ , BYTE ],
;  432          0810  2         ACCOUNT  : REF VECTOR [ , BYTE ];
;  433          0811  2
;  434          0812  2     CH$MOVE ( VVP$S_USERNAME, .USERNAME, CTL$T_USERNAME);
;  435          0813  2     CH$MOVE ( VVP$S_ACCOUNT,  .ACCOUNT,  CTL$T_ACCOUNT);
;  436          0814  1     END;


                                                003C 00000 FILL_PROC:
                                                           .WORD   Save R2,R3,R4,R5                                ; 0773
            00000000G  9F       04   BC             0C  28 00002       MOVC3   #12, @USERNAME, @#CTL$T_USERNAME    ; 0812
            00000000G  9F       08   BC             08  28 0000B       MOVC3   #8, @ACCOUNT, @#CTL$T_ACCOUNT       ; 0813
                                                       04 00014       RET                                         ; 0814

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 0169

;  437          0815  1 END
```

```
;  438          0816  1
;  439          0817  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $LOCKEDD1$                     128  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         382  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                 Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        26         0        1000        00:01.8



;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:OPRCOM/OBJ=OBJ$:OPRCOM MSRC$:OPRCOM/UPDATE=(ENH$:OPRCOM)

;  440          0818  0
; Size:         374 code + 136 data bytes
; Run Time:       00:11.0
; Elapsed Time:   00:23.4
; Lines/CPU Min:   4453
; Lexemes/CPU-Min: 19655
; Memory Used:  105 pages
; Compilation Complete
```

LOCKDB LIS

LOGIO LIS

MATCHNAME LIS

IODONE LIS

LOCKDN LIS

NXTVOL LIS

MAIL LIS

MOUVOL LIS

OPRCOM LIS

INIMTA LIS

MODIFY LIS

PARSE LIS