```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCCC  KKK         KKK  UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCCC  KKK         KKK  UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCCC  KKK         KKK  UUU         UUU  PPPPPPPPPPPP
BBB         BBB  AAA         AAA  CCC             KKK         KKK  UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC             KKK         KKK  UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC             KKK         KKK  UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC             KKK      KKK     UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC             KKK      KKK     UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC             KKK      KKK     UUU         UUU  PPP         PPP
BBBBBBBBBBBB     AAA         AAA  CCC             KKKKKKKKK        UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC             KKKKKKKKK        UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC             KKKKKKKKK        UUU         UUU  PPPPPPPPPPPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC             KKK      KKK     UUU         UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC             KKK      KKK     UUU         UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC             KKK      KKK     UUU         UUU  PPP
BBB         BBB  AAA         AAA  CCC             KKK         KKK  UUU         UUU  PPP
BBB         BBB  AAA         AAA  CCC             KKK         KKK  UUU         UUU  PPP
BBB         BBB  AAA         AAA  CCC             KKK         KKK  UUU         UUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCCC  KKK         KKK  UUUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCCC  KKK         KKK  UUUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCCC  KKK         KKK  UUUUUUUUUUUUUUUU  PPP
```

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA      IIIIII      OOOOOO      SSSSSSSS  YY      YY    SSSSSSSS
  SSSSSSSS  TTTTTTTTTT    AAAAAA      IIIIII      OOOOOO      SSSSSSSS  YY      YY    SSSSSSSS
SS              TT      AA      AA      II      OO      OO  SS          YY      YY  SS
SS              TT      AA      AA      II      OO      OO  SS          YY      YY  SS
SS              TT      AA      AA      II      OO      OO  SS            YY  YY    SS
SS              TT      AA      AA      II      OO      OO  SS            YY  YY    SS
  SSSSSS        TT      AA      AA      II      OO      OO    SSSSSS        YY        SSSSSS
  SSSSSS        TT      AA      AA      II      OO      OO    SSSSSS        YY        SSSSSS
        SS      TT      AAAAAAAAAA      II      OO      OO          SS      YY              SS
        SS      TT      AAAAAAAAAA      II      OO      OO          SS      YY              SS
        SS      TT      AA      AA      II      OO      OO          SS      YY              SS      ....
        SS      TT      AA      AA      II      OO      OO          SS      YY              SS      ....
SSSSSSSS        TT      AA      AA    IIIIII      OOOOOO    SSSSSSSS        YY      SSSSSSSS        ....
SSSSSSSS        TT      AA      AA    IIIIII      OOOOOO    SSSSSSSS        YY      SSSSSSSS        ....


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE STAIOSYS (%TITLE 'Standalone I/O system'
0                  IDENT = 'V04-000'
0                  ) =
1 BEGIN
1
1 !
1 !*****************************************************************************
1 !*                                                                           *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
1 !*  ALL RIGHTS RESERVED.                                                     *
1 !*                                                                           *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
1 !*  TRANSFERRED.                                                             *
1 !*                                                                           *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
1 !*  CORPORATION.                                                             *
1 !*                                                                           *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
1 !*                                                                           *
1 !*                                                                           *
1 !*****************************************************************************
1 !
1
1 !++
1 ! FACILITY:
1 !       Backup/Restore
1 !
1 ! ABSTRACT:
1 !       This module contains the replacements for I/O routines in the
1 !       stand-alone environment.
1 !
1 ! ENVIRONMENT:
1 !       VAX/VMS user mode.
1 !--
1 !
1 ! AUTHOR: M. Jack, CREATION DATE: 27-Dec-1980
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-001 ACG0313         Andrew C. Goldstein,    12-Feb-1983  17:22
1 !               Add routine subtitles
1 !
1 !**
```

```
;   52          0051  1 REQUIRE 'SRC$:COMMON';
;   53          1157  1 LIBRARY 'SYS$LIBRARY:LIB';
;   54          1158  1
;   55          1159  1
;   56          1160  1 FORWARD ROUTINE
;   57          1161  1         ASSIGN_CHANNEL,                 ! Execute $ASSIGN service
;   58          1162  1         LIB$GET_COMMAND,                ! LIB$GET_COMMAND
;   59          1163  1         LIB$PUT_OUTPUT,                 ! LIB$PUT_OUTPUT
;   60          1164  1         SYS$QIO,                        ! QIO service
;   61          1165  1         SYS$QIOW,                       ! QIOW service
;   62          1166  1         SYS$CLOSE,                      ! CLOSE service
;   63          1167  1         SYS$CREATE,                     ! CREATE service
;   64          1168  1         SYS$OPEN,                       ! OPEN service
;   65          1169  1         SYS$CONNECT,                    ! CONNECT service
;   66          1170  1         SYS$PUT,                        ! PUT service
;   67          1171  1         DUMMY;                          ! Dummy service
;   68          1172  1
;   69          1173  1
;   70          1174  1 EXTERNAL ROUTINE
;   71          1175  1         STA_QIO;                        ! Standalone ACP QIO processor
;   72          1176  1
;   73          1177  1
;   74          1178  1 OWN
;   75          1179  1         STA_TT_CHAN;                    ! Channel assigned to SYS$INPUT/OUTPUT
```

```
1 %SBTTL 'ASSIGN_CHANNEL - assign channel to device'
1 ROUTINE ASSIGN_CHANNEL(DEVICE,CHANNEL)=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine executes the $ASSIGN service.
1 !
1 ! INPUT PARAMETERS:
1 !       DEVICE          - Descriptor for device name
1 !       CHANNEL         - Pointer to where channel number will be stored
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       Completion status.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 MAP
2         DEVICE:         REF BBLOCK;     ! Pointer to descriptor for device
2 LOCAL
2         IN_DESC:        VECTOR[2],      ! Descriptor for input device name
2         OUT_DESC:       VECTOR[2],      ! Descriptor for BUFFER
2         BUFFER:         VECTOR[64,BYTE],! Buffer for result of $TRNLOG
2         P,                              ! Temporary pointer
2         STATUS;                         ! Status variable
2
2
2 ! Strip elements following colon from string if required.
2 !
2 IN_DESC[0] = .DEVICE[DSC$W_LENGTH];
2 IF (P = CH$FIND_CH(.DEVICE[DSC$W_LENGTH], .DEVICE[DSC$A_POINTER], %C':')) NEQ 0
2     THEN IN_DESC[0] = .P - .DEVICE[DSC$A_POINTER];
2 IN_DESC[1] = .DEVICE[DSC$A_POINTER];
2
2
2 ! Execute $TRNLOG service to handle process-permanent file.
2 !
2 OUT_DESC[0] = 64;
2 OUT_DESC[1] = BUFFER;
2 BUFFER[0] = 0;
2 $TRNLOG(LOGNAM=IN_DESC, RSLLEN=OUT_DESC, RSLBUF=OUT_DESC);
2
2
2 ! Handle process-permanent file.
```

```
;  134          1237  2 !
;  135          1238  2 IF .BUFFER[0] EQL %O'033'
;  136          1239  2 THEN
;  137          1240  3     BEGIN
;  138          1241  3     IN_DESC[0] = .OUT_DESC[0] - 4;
;  139          1242  3     IN_DESC[1] = .OUT_DESC[1] + 4;
;  140          1243  2     END;
;  141          1244  2
;  142          1245  2
;  143          1246  2 ! Execute $ASSIGN service, returning its result as value.
;  144          1247  2 !
;  145          1248  3 $ASSIGN(DEVNAM=IN_DESC, CHAN=.CHANNEL)
;  146          1249  1 END;


                                                        .TITLE  STAIOSYS Standalone I/O system
                                                        .IDENT  \V04-000\

                                                        .PSECT  DATA,NOEXE,2

                                                00000 STA_TT_CHAN:
                                                        .BLKB   4

                                                        .EXTRN  STA_QIO, SYS$TRNLOG
                                                        .EXTRN  SYS$ASSIGN

                                                        .PSECT  CODE,NOWRT,2

                                                0004 00000 ASSIGN_CHANNEL:
                                                        .WORD   Save R2                                  ; 1181
                        5E      B0   AE  9E 00002       MOVAB   -80(SP), SP                              ;
                        52      04   AC  D0 00006       MOVL    DEVICE, R2                               ; 1222
                48      AE           62  3C 0000A       MOVZWL  (R2), IN_DESC                            ;
04   B2                 62           3A  3A 0000E       LOCC    #58, (R2), @4(R2)                        ; 1223
                                     02  12 00013       BNEQ    1$                                       ;
                                     51  D4 00015       CLRL    R1                                       ;
                                     51  D5 00017 1$:   TSTL    P                                        ;
                                     06  13 00019       BEQL    2$                                       ;
48   AE                 51      04   A2  C3 0001B       SUBL3   4(R2), P, IN_DESC                        ; 1224
                4C      AE      04   A2  D0 00021 2$:   MOVL    4(R2)  N_DESC+4                          ; 1225
                40      AE      40   8F  9A 00026       MOVZBL  #64, OUT_DESC                            ; 1230
                44      AE           6E  9E 0002B       MOVAB   BUFFER, OUT_DESC+4                       ; 1231
                                     6E  94 0002F       CLRB    BUFFER                                   ; 1232
                                     7E  7C 00031       CLRQ    -(SP)                                    ; 1233
                                     7E  D4 00033       CLRL    -(SP)                                    ;
                                4C   AE  9F 00035       PUSHAB  OUT_DESC                                 ;
                                50   AE  9F 00038       PUSHAB  OUT_DESC                                 ;
                                5C   AE  9F 0003B       PUSHAB  IN_DESC                                  ;
            00000000G   00           06  FB 0003E       CALLS   #6, SYS$TRNLOG                           ;
                        1B           6E  91 00045       CMPB    BUFFER, #27                              ; 1238
                                     0C  12 00048       BNEQ    3$                                       ;
48   AE          40      AE           04  C3 0004A       SUBL3   #4, OUT_DESC, IN_DESC                    ; 1241
4C   AE          44      AE           04  C1 00050       ADDL3   #4, OUT_DESC+4, IN_DESC+4                ; 1242
                                     7E  7C 00056 3$:   CLRQ    -(SP)                                    ; 1248
                                08   AC  DD 00058       PUSHL   CHANNEL                                  ;
                                54   AE  9F 0005B       PUSHAB  IN_DESC                                  ;
            00000000G   00           04  FB 0005E       CALLS   #4, SYS$ASSIGN                           ;
```

```
                                        04 00065        RET                                                     ; 1249

; Routine Size:  102 bytes,    Routine Base:  CODE + 0000
```

```
%SBTTL 'LIB$GET_COMMAND - get input from terminal'
GLOBAL ROUTINE LIB$GET_COMMAND(GET_STR,PROMPT_STR,OUT_LEN)=

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine is a replacement for LIB$GET_COMMAND in the
!       stand-alone environment.  Note that it need only handle static
!       output strings.
!
! INPUT PARAMETERS:
!       GET_STR         - Pointer to static descriptor to receive the string.
!       PROMPT_STR      - Pointer to descriptor for prompt string.
!       OUT_LEN         - (Optional) Pointer to word to receive length.
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       Completion status.
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
MAP
        GET_STR:        REF BBLOCK,             ! Pointer to static descriptor
        PROMPT_STR:     REF BBLOCK;             ! Pointer to descriptor
LOCAL
        STATUS,                                 ! Status variable
        IOSB:           VECTOR[4,WORD],         ! I/O status block
        BUFFER:         VECTOR[132,BYTE];       ! Input buffer
BUILTIN
        ACTUALCOUNT;

! Assign the channel if required.
!
IF .STA_TT_CHAN EQL 0
THEN
    BEGIN
    STATUS = ASSIGN_CHANNEL($DESCRIPTOR('SYS$INPUT'), STA_TT_CHAN);
    IF NOT .STATUS THEN RETURN .STATUS;
    END;

! Execute the QIO.
!
STATUS = $QIOW(
```

```
;  205        P 1307  2      FUNC=IO$_READPROMPT,
;  206        P 1308  2      CHAN=.STA_TT_CHAN,
;  207        P 1309  2      IOSB=IOSB,
;  208        P 1310  2      P1=BUFFER,
;  209        P 1311  2      P2=132,
;  210        P 1312  2      P5=.PROMPT_STR[DSC$A_POINTER],
;  211          1313  2      P6=.PROMPT_STR[DSC$W_LENGTH]);
;  212          1314  2 IF .STATUS THEN STATUS = .IOSB[0];
;  213          1315  2 IF NOT .STATUS THEN RETURN .STATUS;
;  214          1316  2
;  215          1317  2
;  216          1318  2 ! Return the length if requested.
;  217          1319  2 !
;  218          1320  2 IF ACTUALCOUNT() GEQU 3
;  219          1321  2 THEN
;  220          1322  2     (.OUT_LEN)<0,16> = MINU(.IOSB[1], .GET_STR[DSC$W_LENGTH]);
;  221          1323  2
;  222          1324  2
;  223          1325  2 ! Copy the string to the output buffer and return.
;  224          1326  2 !
;  225          1327  2 CH$COPY(.IOSB[1], BUFFER, %C' ', .GET_STR[DSC$W_LENGTH], .GET_STR[DSC$A_POINTER]);
;  226          1328  2 SS$_NORMAL
;  227          1329  1 END;


                    54  55  50  4E  49  24  53  59  53  00066 P.AAB:  .ASCII  \SYS$INPUT\                          ;
                                                        0006F         .BLKB   1
                                                 00000009  00070 P.AAA:  .LONG   9                                  ;
                                                 00000000' 00074         .ADDRESS P.AAB                             ;

                                                                         .EXTRN  SYS$QIOW

                                                     007C  00000         .ENTRY  LIB$GET_COMMAND, Save R2,R3,R4,R5,R6  ; 1251
                        56 00000000'  EF   9E  00002         MOVAB   STA_TT_CHAN, R6                                ;
                        5E      FF74  CE   9E  00009         MOVAB   -140(SP), SP                                   ;
                                      66   D5  0000E         TSTL    STA_TT_CHAN                                    ; 1296
                                      0D   12  00010         BNEQ    1$                                             ;
                                      56   DD  00012         PUSHL   R6                                             ; 1299
                                 E1   AF   9F  00014         PUSHAB  P.AAA                                          ;
                 FF6C   CF            02   FB  00017         CALLS   #2, ASSIGN_CHANNEL                             ;
                        55            50   E9  0001C         BLBC    STATUS, 4$                                     ; 1300
                        51       08   AC   D0  0001F  1$:     MOVL    PROMPT_STR, R1                                 ; 1313
                        7E            61   3C  00023         MOVZWL  (R1), -(SP)                                    ;
                                 04   A1   DD  00026         PUSHL   4(R1)                                          ;
                                      7E   7C  00029         CLRQ    -(SP)                                          ;
                        7E       84   8F   9A  0002B         MOVZBL  #132, -(SP)                                    ;
                                 14   AE   9F  0002F         PUSHAB  BUFFER                                         ;
                                      7E   7C  00032         CLRQ    -(SP)                                          ;
                                 F8   AD   9F  00034         PUSHAB  IOSB                                           ;
                                      37   DD  00037         PUSHL   #55                                            ;
                                      66   DD  00039         PUSHL   STA_TT_CHAN                                    ;
                                      7E   D4  0003B         CLRL    -(SP)                                          ;
                 00000000G   00       0C   FB  0003D         CALLS   #12, SYS$QIOW                                  ;
                             2D       50   E9  00044         BLBC    STATUS, 4$                                     ; 1314
                             50       F8   AD  3C  00047     MOVZWL  IOSB, STATUS                                   ;
                             26            50  E9  0004B     BLBC    STATUS, 4$                                     ; 1315
```

```
                        03          6C  91 0004E          CMPB    (AP), #3                                  ; 1320
                                    12  1F 00051          BLSSU   3$                                        ;
                        50      FA  AD  3C 00053          MOVZWL  IOSB+2, R0                                ; 1322
                        50      04  BC  B1 00057          CMPW    @GET_STR, R0                              ;
                                    04  1E 0005B          BGEQU   2$                                        ;
                        50      04  BC  3C 0005D          MOVZWL  @GET_STR, R0                              ;
                    0C  BC          50  B0 00061 2$:      MOVW    R0, @OUT_LEN                              ;
                        50      04  AC  D0 00065 3$:      MOVL    GET_STR, R0                               ; 1327
        60          20  6E      FA  AD  2C 00069          MOVC5   IOSB+2, BUFFER, #32, (R0), @4(R0)         ;
                            04  B0      0006F                                                               ;
                        50          01  D0 00071          MOVL    #1, R0                                    ; 1329
                                        04 00074 4$:      RET                                               ;

; Routine Size:  117 bytes,    Routine Base:  CODE + 0078
```

```
%SBTTL 'LIB$PUT_OUTPUT - write to SYS$OUTPUT'
GLOBAL ROUTINE LIB$PUT_OUTPUT(PUT_STR)=

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine is a replacement for LIB$PUT_OUTPUT in the
!       stand-alone environment.
!
! INPUT PARAMETERS:
!       PUT_STR         - Pointer to descriptor for string.
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       Completion status.
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
MAP
        PUT_STR:        REF BBLOCK;     ! Pointer to descriptor
LOCAL
        STATUS,                         ! Status variable
        IOSB:           VECTOR[4,WORD]; ! I/O status block


! Assign the channel if required.
!
IF .STA_TT_CHAN EQL 0
THEN
    BEGIN
    STATUS = ASSIGN_CHANNEL($DESCRIPTOR('SYS$INPUT'), STA_TT_CHAN);
    IF NOT .STATUS THEN RETURN .STATUS;
    END;


! Execute the QIO.
!
STATUS = $QIOW(
    FUNC=IO$_WRITELBLK,
    CHAN=.STA_TT_CHAN,
    IOSB=IOSB,
    P1=.PUT_STR[DSC$A_POINTER],
    P2=.PUT_STR[DSC$W_LENGTH],
    P4=%C' ');
IF .STATUS THEN STATUS = .IOSB[0];
```

```
;  286          1387  2 .STATUS
;  287          1388  1 END;


                54  55  50  4E  49  24  53  59  53  000ED P.AAD:  .ASCII  \SYS$INPUT\                          ;
                                                    000F6         .BLKB   2
                                            00000009 000F8 P.AAC:  .LONG   9                                  ;
                                            00000000' 000FC        .ADDRESS P.AAD                             ;


                                                0004 00000        .ENTRY  LIB$PUT_OUTPUT, Save R2            ; 1331
                        52 00000000'  EF  9E 00002        MOVAB   STA_TT_CHAN, R2                            ;
                        5E            08  C2 00009        SUBL2   #8, SP                                     ;
                                      62  D5 0000C        TSTL    STA_TT_CHAN                                ; 1369
                                      0D  12 0000E        BNEQ    1$                                         ;
                                      52  DD 00010        PUSHL   R2                                         ; 1372
                                 E3   AF  9F 00012        PUSHAB  P.AAC                                      ;
                    FEE6        CF    02  FB 00015        CALLS   #2, ASSIGN_CHANNEL                         ;
                                28    50  E9 0001A        BLBC    STATUS, 2$                                 ; 1373
                                      7E  7C 0001D 1$:    CLRQ    -(SP)                                      ; 1385
                                      20  DD 0001F        PUSHL   #32                                        ;
                                      7E  D4 00021        CLRL    -(SP)                                      ;
                        51       04   AC  D0 00023        MOVL    PUT_STR, R1                                ;
                        7E            61  3C 00027        MOVZWL  (R1), -(SP)                                ;
                                 04   A1  DD 0002A        PUSHL   4(R1)                                      ;
                                      7E  7C 0002D        CLRQ    -(SP)                                      ;
                                 20   AE  9F 0002F        PUSHAB  IOSB                                       ;
                                      20  DD 00032        PUSHL   #32                                        ;
                                      62  DD 00034        PUSHL   STA_TT_CHAN                                ;
                                      7E  D4 00036        CLRL    -(SP)                                      ;
                    00000000G   00    0C  FB 00038        CALLS   #12, SYS$QIOW                              ;
                                03    50  E9 0003F        BLBC    STATUS, 2$                                 ; 1386
                                50    6E  3C 00042        MOVZWL  IOSB, STATUS                               ;
                                            04 00045 2$:  RET                                                ; 1388

; Routine Size:  70 bytes,    Routine Base:  CODE + 0100
```

```
%SBTTL 'SYS$QIO - intercept QIO service call'
GLOBAL ROUTINE SYS$QIO(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine is a replacement for the QIO service in the
!       stand-alone environment.
!
! INPUT PARAMETERS:
!       As for SYS$QIO.
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       Completion status.
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
MAP
        FUNC:           BBLOCK;         ! I/O function code
BIND ROUTINE
        QIO_VECTOR=     %X'800001C8';   ! Vector location for SYS$QIO
BUILTIN
        CALLG,
        AP;


IF
    .FUNC[IO$V_FCODE] LEQU IO$_LOGICAL OR
    .FUNC[IO$V_FCODE] EQL IO$_READPROMPT
THEN
    BEGIN
    !
    ! Pass physical and logical I/O functions, and READPROMPT function,
    ! directly to the driver.
    !
    CALLG(.AP, QIO_VECTOR)
    END
ELSE
    BEGIN
    !
    ! Read/write virtual and ACP I/O functions come here.
    !
    CALLG(.AP, STA_QIO)
    END
```

```
;  346          1446  1 END;


                                                    QIO_VECTOR=          -2147483192


                                          0000 00000          .ENTRY  SYS$QIO, Save nothing                    ; 1390
        2F      0C   AC          06     00  ED 00002          CMPZV   #0, #6, FUNC, #47                        ; 1429
                                        08  1B 00008          BLEQU   1$                                       ;
        37      0C   AC          06     00  ED 0000A          CMPZV   #0, #6, FUNC, #55                        ; 1430
                                        08  12 00010          BNEQ    2$                                       ;
                     800001C8    9F     6C  FA 00012 1$:      CALLG   (AP), @#^X800001C8                       ; 1437
                                        04 00019              RET                                              ; 1432
                     00000000G   00     6C  FA 0001A 2$:      CALLG   (AP), STA_QIO                            ; 1444
                                        04 00021              RET                                              ; 1446

; Routine Size:  34 bytes,    Routine Base:  CODE + 0146
```

```
;  348          1447  1 %SBTTL 'SYS$QIOW - intercept QIOW service call'
;  349          1448  1 GLOBAL ROUTINE SYS$QIOW(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  350          1449  1
;  351          1450  1 !++
;  352          1451  1 !
;  353          1452  1 ! FUNCTIONAL DESCRIPTION:
;  354          1453  1 !       This routine is a replacement for the QIOW service in the
;  355          1454  1 !       stand-alone environment.
;  356          1455  1 !
;  357          1456  1 ! INPUT PARAMETERS:
;  358          1457  1 !       As for SYS$QIOW.
;  359          1458  1 !
;  360          1459  1 ! IMPLICIT INPUTS:
;  361          1460  1 !       NONE
;  362          1461  1 !
;  363          1462  1 ! OUTPUT PARAMETERS:
;  364          1463  1 !       NONE
;  365          1464  1 !
;  366          1465  1 ! IMPLICIT OUTPUTS:
;  367          1466  1 !       NONE
;  368          1467  1 !
;  369          1468  1 ! ROUTINE VALUE:
;  370          1469  1 !       Completion status.
;  371          1470  1 !
;  372          1471  1 ! SIDE EFFECTS:
;  373          1472  1 !       NONE
;  374          1473  1 !
;  375          1474  1 !--
;  376          1475  1
;  377          1476  2 BEGIN
;  378          1477  2 EXTERNAL ROUTINE
;  379          1478  2         SYS$WAITFR:     ADDRESSING_MODE(GENERAL);       ! Wait for event flag
;  380          1479  2 REGISTER
;  381          1480  2         R0=0;
;  382          1481  2 BUILTIN
;  383          1482  2         CALLG,
;  384          1483  2         AP;
;  385          1484  2
;  386          1485  2
;  387          1486  2 R0 = CALLG(.AP, SYS$QIO);       ! Execute $QIO part
;  388          1487  2 IF NOT .R0 THEN RETURN .R0;     ! If failed, return status
;  389          1488  2 CALLG(.AP, SYS$WAITFR)          ! Execute $WAITFR part
;  390          1489  1 END;


                                                        .EXTRN  SYS$WAITFR

                                        0000 00000      .ENTRY  SYS$QIOW, Save nothing          ; 1448
                    D8   AF         6C  FA 00002        CALLG   (AP), SYS$QIO                   ; 1486
                         07         50  E9 00006        BLBC    R0, 1$                          ; 1487
            00000000G    00         6C  FA 00009        CALLG   (AP), SYS$WAITFR                ; 1488
                                        04 00010 1$:    RET                                     ; 1489

; Routine Size:  17 bytes,    Routine Base:  CODE + 0168
```

```
;  392          1490  1 %SBTTL 'SYS$CLOSE - RMS $CLOSE routine'
;  393          1491  1 GLOBAL ROUTINE SYS$CLOSE(FAB)=
;  394          1492  1
;  395          1493  1 !++
;  396          1494  1 !
;  397          1495  1 ! FUNCTIONAL DESCRIPTION:
;  398          1496  1 !       This routine is a replacement for the CLOSE service in the
;  399          1497  1 !       stand-alone environment.
;  400          1498  1 !
;  401          1499  1 ! INPUT PARAMETERS:
;  402          1500  1 !       FAB             - Pointer to the FAB.
;  403          1501  1 !
;  404          1502  1 ! IMPLICIT INPUTS:
;  405          1503  1 !       NONE
;  406          1504  1 !
;  407          1505  1 ! OUTPUT PARAMETERS:
;  408          1506  1 !       NONE
;  409          1507  1 !
;  410          1508  1 ! IMPLICIT OUTPUTS:
;  411          1509  1 !       NONE
;  412          1510  1 !
;  413          1511  1 ! ROUTINE VALUE:
;  414          1512  1 !       Completion status.
;  415          1513  1 !
;  416          1514  1 ! SIDE EFFECTS:
;  417          1515  1 !       NONE
;  418          1516  1 !
;  419          1517  1 !--
;  420          1518  1
;  421          1519  2 BEGIN
;  422          1520  2 MAP
;  423          1521  2         FAB:            REF BBLOCK;     ! Pointer to FAB
;  424          1522  2
;  425          1523  2
;  426          1524  2 ! This service deassigns the channel that was assigned by $CREATE or $OPEN.
;  427          1525  2 !
;  428          1526  2 $DASSGN(CHAN=.FAB[FAB$L_CTX]);
;  429          1527  2 FAB[FAB$W_IFI] = 0;
;  430          1528  2 FAB[FAB$L_CTX] = 0;
;  431          1529  2 FAB[FAB$L_STS] = RMS$_NORMAL;
;  432          1530  2 FAB[FAB$L_STV] = 0;
;  433          1531  2 RMS$_NORMAL
;  434          1532  1 END;


                                                        .EXTRN  SYS$DASSGN

                                        0004 00000      .ENTRY  SYS$CLOSE, Save R2                      ; 1491
                 52        04   AC   D0 00002          MOVL    FAB, R2                                 ; 1526
                           18   A2   DD 00006          PUSHL   24(R2)                                  ;
        00000000G  00             01   FB 00009         CALLS   #1, SYS$DASSGN                          ;
                           02   A2   B4 00010          CLRW    2(R2)                                   ; 1527
                           18   A2   D4 00013          CLRL    24(R2)                                  ; 1528
             08    A2 00010001   8F   D0 00016          MOVL    #65537, 8(R2)                           ; 1529
                           0C   A2   D4 0001E          CLRL    12(R2)                                  ; 1530
                 50 00010001   8F   D0 00021          MOVL    #65537, R0                              ; 1532
```

```
                                        04 00028        RET                                                     ;

; Routine Size:  41 bytes,    Routine Base:  CODE + 0179
```

```
%SBTTL 'SYS$CREATE - RMS $CREATE routine'
GLOBAL ROUTINE SYS$CREATE(FAB)=

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine is a replacement for the CREATE service in the
!       stand-alone environment.  It is limited to operation on terminals,
!       line printers, and magnetic tape.
!
! INPUT PARAMETERS:
!       FAB             - Pointer to the FAB.
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       Completion status.
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
MAP
        FAB:            REF BBLOCK;     ! Pointer to FAB
LOCAL
        NAM:            REF BBLOCK,     ! Pointer to NAM block
        DESC:           VECTOR[2],      ! Descriptor for expanded string
        STATUS;                         ! Status variable


! Parse the file specification if necessary.
!
IF NOT $PARSE(FAB=.FAB) THEN RETURN .FAB[FAB$L_STS];


! Check the device type.
!
IF
    NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_CCL] AND
    NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD] AND
    .FAB[FAB$L_DEV] NEQ 0 ! Process-permanent file hack
THEN
    BEGIN
    FAB[FAB$L_STS] = RMS$_DEV;
    FAB[FAB$L_STV] = 0;
    RETURN RMS$_DEV
    END;
```

```
;  493          1590  2
;  494          1591  2 ! Create the resultant string.
;  495          1592  2 !
;  496          1593  2 NAM = .FAB[FAB$L_NAM];
;  497          1594  2 NAM[NAM$B_RSL] = .NAM[NAM$B_ESL];
;  498          1595  2 CH$MOVE(.NAM[NAM$B_ESL], .NAM[NAM$L_ESA], .NAM[NAM$L_RSA]);
;  499          1596  2
;  500          1597  2
;  501          1598  2 ! Assign the channel.
;  502          1599  2 !
;  503          1600  2 DESC[0] = .NAM[NAM$B_ESL];
;  504          1601  2 DESC[1] = .NAM[NAM$L_ESA];
;  505          1602  2 STATUS = ASSIGN_CHANNEL(DESC, FAB[FAB$L_CTX]);
;  506          1603  2 IF NOT .STATUS
;  507          1604  2 THEN
;  508          1605  3     BEGIN
;  509          1606  3     FAB[FAB$L_STS] = .STATUS;
;  510          1607  3     FAB[FAB$L_STV] = 0;
;  511          1608  3     RETURN .STATUS;
;  512          1609  2     END;
;  513          1610  2
;  514          1611  2
;  515          1612  2 ! Indicate success.
;  516          1613  2 !
;  517          1614  2 FAB[FAB$L_STS] = RMS$_NORMAL;
;  518          1615  2 FAB[FAB$L_STV] = .FAB[FAB$L_CTX];
;  519          1616  2 RMS$_NORMAL
;  520          1617  1 END;


                                                        .EXTRN  SYS$PARSE

                                     00FC 00000         .ENTRY  SYS$CREATE, Save R2,R3,R4,R5,R6,R7      ; 1534
                        5E             08  C2 00002     SUBL2   #8, SP                                  ;
                        57       04    AC  D0 00005     MOVL    FAB, R7                                 ; 1574
                                       57  DD 00009     PUSHL   R7                                      ;
            00000000G   00             01  FB 0000B     CALLS   #1, SYS$PARSE                           ;
                        05             50  E8 00012     BLBS    R0, 1$                                  ;
                        50       08    A7  D0 00015     MOVL    8(R7), R0                               ;
                                       04  00019         RET                                             ;
        1D      40      A7             01  E0 0001A 1$:  BBS     #1, 64(R7), 2$                          ; 1580
        18      40      A7             05  E0 0001F     BBS     #5, 64(R7), 2$                          ; 1581
                                 40    A7  D5 00024     TSTL    64(R7)                                  ; 1582
                                       13  13 00027     BEQL    2$                                      ;
                40      A7 000184C4    8F  D0 00029     MOVL    #99524, 8(R7)                           ; 1585
                                   0C  A7  D4 00031     CLRL    12(R7)                                  ; 1586
                        50 000184C4    8F  D0 00034     MOVL    #99524, R0                              ; 1587
                                       04  0003B         RET                                             ;
                        56       28    A7  D0 0003C 2$:  MOVL    40(R7), NAM                             ; 1593
                03      A6       0B    A6  90 00040     MOVB    11(NAM), 3(NAM)                         ; 1594
                        50       0B    A6  9A 00045     MOVZBL  11(NAM), R0                             ; 1595
    04  B6      0C      B6             50  28 00049     MOVC3   R0, @12(NAM), @4(NAM)                   ;
                        6E       0B    A6  9A 0004F     MOVZBL  11(NAM), DESC                           ; 1600
                04      AE       0C    A6  D0 00053     MOVL    12(NAM), DESC+4                         ; 1601
                                 18    A7  9F 00058     PUSHAB  24(R7)                                  ; 1602
                                 04    AE  9F 0005B     PUSHAB  DESC                                    ;
```

```
                FDFB   CF            02  FB 0005E          CALLS   #2, ASSIGN_CHANNEL                  ;
                       08            50  E8 00063          BLBS    STATUS, 3$                          ; 1603
                   08  A7            50  D0 00066          MOVL    STATUS, 8(R7)                       ; 1606
                                  0C  A7  D4 0006A          CLRL    12(R7)                              ; 1607
                                          04 0006D          RET                                         ; 1608
                   08  A7 00010001   8F  D0 0006E 3$:       MOVL    #65537, 8(R7)                       ; 1614
                   0C  A7        18  A7  D0 00076          MOVL    24(R7), 12(R7)                      ; 1615
                       50 00C10001   8F  D0 0007B          MOVL    #65537, R0                          ; 1617
                                          04 00082          RET                                         ;

; Routine Size:  131 bytes,    Routine Base:  CODE + 01A2
```

```
%SBTTL 'SYS$OPEN - RMS $OPEN routine'
GLOBAL ROUTINE SYS$OPEN(FAB)=

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine is a replacement for the OPEN service in the
!       stand-alone environment.  It is limited to operation on terminals,
!       line printers, and magnetic tape.
!
! INPUT PARAMETERS:
!       FAB             - Pointer to the FAB.
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       Completion status.
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
MAP
        FAB:            REF BBLOCK;     ! Pointer to FAB
LOCAL
        NAM:            REF BBLOCK,     ! Pointer to NAM block
        DESC:           VECTOR[2],      ! Descriptor for expanded string
        STATUS;                         ! Status variable


! Parse the file specification if necessary.
!
IF NOT $PARSE(FAB=.FAB) THEN RETURN .FAB[FAB$L_STS];


! Check the device type.
!
IF
    NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_CCL] AND
    NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD] AND
    .FAB[FAB$L_DEV] NEQ 0 ! Process-permanent file hack
THEN
    BEGIN
    FAB[FAB$L_STS] = RMS$_DEV;
    FAB[FAB$L_STV] = 0;
    RETURN RMS$_DEV
    END;
```

```
;  579          1675  2
;  580          1676  2 ! Create the resultant string.
;  581          1677  2 !
;  582          1678  2 NAM = .FAB[FAB$L_NAM];
;  583          1679  2 NAM[NAM$B_RSL] = .NAM[NAM$B_ESL];
;  584          1680  2 CH$MOVE(.NAM[NAM$B_ESL], .NAM[NAM$L_ESA], .NAM[NAM$L_RSA]);
;  585          1681  2
;  586          1682  2
;  587          1683  2 ! Assign the channel.
;  588          1684  2 !
;  589          1685  2 DESC[0] = .NAM[NAM$B_ESL];
;  590          1686  2 DESC[1] = .NAM[NAM$L_ESA];
;  591          1687  2 STATUS = ASSIGN_CHANNEL(DESC, FAB[FAB$L_CTX]);
;  592          1688  2 IF NOT .STATUS
;  593          1689  2 THEN
;  594          1690  3     BEGIN
;  595          1691  3     FAB[FAB$L_STS] = .STATUS;
;  596          1692  3     FAB[FAB$L_STV] = 0;
;  597          1693  3     RETURN .STATUS;
;  598          1694  2     END;
;  599          1695  2
;  600          1696  2
;  601          1697  2 ! Indicate success.
;  602          1698  2 !
;  603          1699  2 FAB[FAB$L_STS] = RMS$_NORMAL;
;  604          1700  2 FAB[FAB$L_STV] = .FAB[FAB$L_CTX];
;  605          1701  2 RMS$_NORMAL
;  606          1702  1 END;


                                  00FC 00000          .ENTRY  SYS$OPEN, Save R2,R3,R4,R5,R6,R7        ; 1619
                 5E                08  C2 00002          SUBL2   #8, SP                                  ;
                 57          04    AC  D0 00005          MOVL    FAB, R7                                 ; 1659
                                   57  DD 00009          PUSHL   R7                                      ;
        00000000G  00              01  FB 0000B          CALLS   #1, SYS$PARSE                           ;
                 05                50  E8 00012          BLBS    R0, 1$                                  ;
                 50          08    A7  D0 00015          MOVL    8(R7), R0                               ;
                                       04 00019          RET                                             ;
     1D      40  A7               01  E0 0001A 1$:      BBS     #1, 64(R7), 2$                          ; 1665
     18      40  A7               05  E0 0001F          BBS     #5, 64(R7), 2$                          ; 1666
                             40    A7  D5 00024          TSTL    64(R7)                                  ; 1667
                                   13  13 00027          BEQL    2$                                      ;
             08  A7 000184C4       8F  D0 00029          MOVL    #99524, 8(R7)                           ; 1670
                             0C    A7  D4 00031          CLRL    12(R7)                                  ; 1671
                 50 000184C4       8F  D0 00034          MOVL    #99524, R0                              ; 1672
                                       04 0003B          RET                                             ;
                 56          28    A7  D0 0003C 2$:      MOVL    40(R7), NAM                             ; 1678
             03  A6          0B    A6  90 00040          MOVB    11(NAM), 3(NAM)                         ; 1679
                 50          0B    A6  9A 00045          MOVZBL  11(NAM), R0                             ; 1680
 04  B6      0C  B6               50  28 00049          MOVC3   R0, @12(NAM), @4(NAM)                   ;
                 6E          0B    A6  9A 0004F          MOVZBL  11(NAM), DESC                           ; 1685
             04  AE          0C    A6  D0 00053          MOVL    12(NAM), DESC+4                         ; 1686
                             18    A7  9F 00058          PUSHAB  24(R7)                                  ; 1687
                             04    AE  9F 0005B          PUSHAB  DESC                                    ;
```

```
                FD78  CF            02  FB 0005E          CALLS   #2, ASSIGN_CHANNEL                  ;
                      08            50  E8 00063          BLBS    STATUS, 3$                          ; 1688
                   08 A7            50  D0 00066          MOVL    STATUS, 8(R7)                       ; 1691
                                 0C A7  D4 0006A          CLRL    12(R7)                              ; 1692
                                        04 0006D          RET                                         ; 1693
                   08 A7 00010001   8F  D0 0006E 3$:      MOVL    #65537, 8(R7)                       ; 1699
                   0C A7       18   A7  D0 00076          MOVL    24(R7), 12(R7)                      ; 1700
                      50 00010001   8F  D0 0007B          MOVL    #65537, R0                          ; 1702
                                        04 00082          RET                                         ;

; Routine Size:  131 bytes,    Routine Base:  CODE + 0225
```

```
;  608          1703  1 %SBTTL 'SYS$CONNECT - RMS $CONNECT routine'
;  609          1704  1 GLOBAL ROUTINE SYS$CONNECT(RAB)=
;  610          1705  1
;  611          1706  1 !++
;  612          1707  1 !
;  613          1708  1 ! FUNCTIONAL DESCRIPTION:
;  614          1709  1 !       This routine is a replacement for the CONNECT service in the
;  615          1710  1 !       stand-alone environment.
;  616          1711  1 !
;  617          1712  1 ! INPUT PARAMETERS:
;  618          1713  1 !       RAB             - Pointer to the RAB.
;  619          1714  1 !
;  620          1715  1 ! IMPLICIT INPUTS:
;  621          1716  1 !       NONE
;  622          1717  1 !
;  623          1718  1 ! OUTPUT PARAMETERS:
;  624          1719  1 !       NONE
;  625          1720  1 !
;  626          1721  1 ! IMPLICIT OUTPUTS:
;  627          1722  1 !       NONE
;  628          1723  1 !
;  629          1724  1 ! ROUTINE VALUE:
;  630          1725  1 !       Completion status.
;  631          1726  1 !
;  632          1727  1 ! SIDE EFFECTS:
;  633          1728  1 !       NONE
;  634          1729  1 !
;  635          1730  1 !--
;  636          1731  1
;  637          1732  2 BEGIN
;  638          1733  2 MAP
;  639          1734  2         RAB:            REF BBLOCK;     ! Pointer to RAB
;  640          1735  2
;  641          1736  2
;  642          1737  2 ! This service is a no-operation.
;  643          1738  2 !
;  644          1739  2 RAB[RAB$L_STS] = RMS$_NORMAL;
;  645          1740  2 RAB[RAB$L_STV] = 0;
;  646          1741  2 RMS$_NORMAL
;  647          1742  1 END;


                                        0000 00000          .ENTRY  SYS$CONNECT, Save nothing                   ; 1704
                                50          04   AC   D0 00002  MOVL    RAB, R0                                 ; 1739
                            08  A0 00010001     8F   D0 00006  MOVL    #65537, 8(R0)                           ;
                                           0C   A0   D4 0000E  CLRL    12(R0)                                  ; 1740
                                50 00010001     8F   D0 00011  MOVL    #65537, R0                              ; 1742
                                                     04 00018  RET                                             ;

; Routine Size:  25 bytes,    Routine Base:  CODE + 02A8
```

```
%SBTTL 'SYS$PUT - RMS $PUT routine'
GLOBAL ROUTINE SYS$PUT(RAB)=

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine is a replacement for the PUT service in the
!       stand-alone environment.  This service is provided to allow
!       /LIST to operate.  It is restricted to terminals and line printers.
!
! INPUT PARAMETERS:
!       RAB             - Pointer to the RAB.
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       Completion status.
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
MAP
        RAB:            REF BBLOCK;     ! Pointer to RAB
LOCAL
        FAB:            REF BBLOCK,     ! Pointer to FAB
        STATUS,                         ! Status variable
        IOSB:           VECTOR[4,WORD]; ! I/O status block


! Point to FAB.
!
FAB = .RAB[RAB$L_FAB];


! Make sure a channel has been assigned.
!
IF .FAB[FAB$L_CTX] EQL 0
THEN
    BEGIN
    RAB[RAB$L_STS] = RMS$_DEV;
    RAB[RAB$L_STV] = 0;
    RETURN RMS$_DEV
    END;


! Execute the I/O.
!
```

```
;  706        P 1800  2 STATUS = $QIOW(
;  707        P 1801  2     FUNC=IO$_WRITELBLK,
;  708        P 1802  2     CHAN=.FAB[FAB$L_CTX],
;  709        P 1803  2     IOSB=IOSB,
;  710        P 1804  2     P1=.RAB[RAB$L_RBF],
;  711        P 1805  2     P2=.RAB[RAB$W_RSZ],
;  712          1806  2     P4=%C' ');
;  713          1807  2 IF .STATUS THEN STATUS = .IOSB[0];
;  714          1808  2 RAB[RAB$L_STS] = .STATUS;
;  715          1809  2 RAB[RAB$L_STV] = 0;
;  716          1810  2 .STATUS
;  717          1811  1 END;


                                    0004 00000          .ENTRY  SYS$PUT, Save R2                    ; 1744
                 5E                 08   C2 00002       SUBL2   #8, SP                              ;
                 52          04     AC   D0 00005       MOVL    RAB, R2                             ; 1784
                 50          3C     A2   D0 00009       MOVL    60(R2), FAB                         ;
                             18     A0   D5 0000D       TSTL    24(FAB)                             ; 1789
                                    13   12 00010       BNEQ    1$                                  ;
            08   A2 000184C4        8F   D0 00012       MOVL    #99524, 8(R2)                       ; 1792
                            0C      A2   D4 0001A       CLRL    12(R2)                              ; 1793
                 50 000184C4        8F   D0 0001D       MOVL    #99524, R0                          ; 1794
                                    04   00024          RET                                         ;
                                    7E   7C 00025 1$:   CLRQ    -(SP)                               ; 1806
                                    20   DD 00027       PUSHL   #32                                 ;
                                    7E   D4 00029       CLRL    -(SP)                               ;
                 7E          22     A2   3C 0002B       MOVZWL  34(R2), -(SP)                       ;
                             28     A2   DD 0002F       PUSHL   40(R2)                              ;
                                    7E   7C 00032       CLRQ    -(SP)                               ;
                             20     AE   9F 00034       PUSHAB  IOSB                                ;
                                    20   DD 00037       PUSHL   #32                                 ;
                             18     A0   DD 00039       PUSHL   24(FAB)                             ;
                                    7E   D4 0003C       CLRL    -(SP)                               ;
       00000000G 00                 0C   FB 0003E       CALLS   #12, SYS$QIOW                       ;
                 03                 50   E9 00045       BLBC    STATUS, 2$                          ; 1807
                 50                 6E   3C 00048       MOVZWL  IOSB, STATUS                        ;
            08   A2                 50   D0 0004B 2$:   MOVL    STATUS, 8(R2)                       ; 1808
                             0C     A2   D4 0004F       CLRL    12(R2)                              ; 1809
                                    04   00052          RET                                         ; 1811

; Routine Size:  83 bytes,    Routine Base:  CODE + 02C1
```

```
;  719          1812  1 %SBTTL 'DUMMY - various dummy RMS routines'
;  720          1813  1 ROUTINE DUMMY(RAB)=
;  721          1814  1
;  722          1815  1 !++
;  723          1816  1 !
;  724          1817  1 ! FUNCTIONAL DESCRIPTION:
;  725          1818  1 !       This routine serves as a stub for services that are referenced but are
;  726          1819  1 !       never called in the stand-alone environment, namely SYS$FIND, SYS$GET,
;  727          1820  1 !       and SYS$REWIND.  These services are used to handle save set files in
;  728          1821  1 !       the on-line version.
;  729          1822  1 !
;  730          1823  1 ! INPUT PARAMETERS:
;  731          1824  1 !       RAB             - Pointer to the RAB.
;  732          1825  1 !
;  733          1826  1 ! IMPLICIT INPUTS:
;  734          1827  1 !       NONE
;  735          1828  1 !
;  736          1829  1 ! OUTPUT PARAMETERS:
;  737          1830  1 !       NONE
;  738          1831  1 !
;  739          1832  1 ! IMPLICIT OUTPUTS:
;  740          1833  1 !       NONE
;  741          1834  1 !
;  742          1835  1 ! ROUTINE VALUE:
;  743          1836  1 !       Completion status (RMS$_IOP, invalid operation).
;  744          1837  1 !
;  745          1838  1 ! SIDE EFFECTS:
;  746          1839  1 !       NONE
;  747          1840  1 !
;  748          1841  1 !--
;  749          1842  1
;  750          1843  2 BEGIN
;  751          1844  2 MAP
;  752          1845  2         RAB:            REF BBLOCK;     ! Pointer to RAB
;  753          1846  2
;  754          1847  2
;  755          1848  2 RAB[RAB$L_STS] = RMS$_IOP;
;  756          1849  2 RAB[RAB$L_STV] = 0;
;  757          1850  2 RMS$_IOP
;  758          1851  1 END;



                                        0000 00000 DUMMY:  .WORD   Save nothing                    ; 1813
                        50        04    AC   D0 00002      MOVL    RAB, R0                         ; 1848
                    08  A0 00018574     8F   D0 00006      MOVL    #99700, 8(R0)                   ;
                                  0C    A0   D4 0000E      CLRL    12(R0)                          ; 1849
                        50 00018574     8F   D0 00011      MOVL    #99700, R0                      ; 1851
                                                04 00018   RET                                     ;

; Routine Size:  25 bytes,    Routine Base:  CODE + 0314
```

```
;  760          1852  1 GLOBAL BIND ROUTINE
;  761          1853  1          SYS$FIND        = DUMMY,
;  762          1854  1          SYS$GET         = DUMMY,
;  763          1855  1          SYS$REWIND      = DUMMY;
;  764          1856  1
;  765          1857  1
;  766          1858  1 END
;  767          1859  0 ELUDOM


                                                    SYS$FIND==          DUMMY
                                                    SYS$GET==           DUMMY
                                                    SYS$REWIND==        DUMMY


;                                PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;  DATA                             4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  CODE                           813  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        36         0      1000          00:02.2




;                                     COMMAND QUALIFIERS

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

; Size:          776 code + 41 data bytes
; Run Time:        00:23.3
; Elapsed Time:    01:11.8
; Lines/CPU Min:    4789
; Lexemes/CPU-Min: 33866
; Memory Used:  185 pages
; Compilation Complete
```

TAPEUTIL
LIS

STAINVOL
LIS

UTILITY
LIS

STAIOSYS
LIS

STAPARSE
LIS