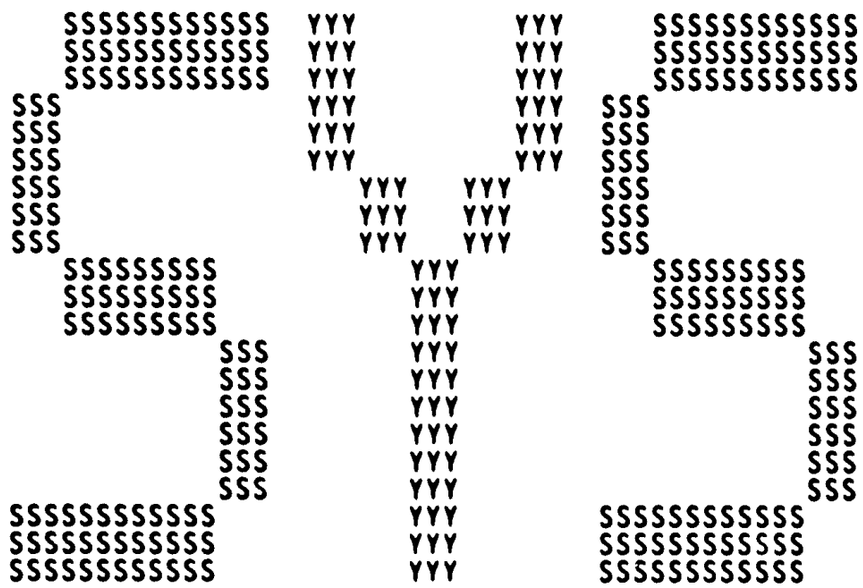

FILEID**SYSDASSGN

Table of contents

```
0000     1          .TITLE  SYSDASSGN - SYSTEM SERVICE DEASSIGN I/O CHANNEL
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  ; D. N. CUTLER 26-AUG-76
0000    29  ;
0000    30  ; MODIFIED BY:
0000    31  ;
0000    32  ;       V03-015 HH0049          Hai Huang               16-Aug-1984
0000    33  ;               Retract HH0048. Device deallocation on dismount will
0000    34  ;               be done in the file systems and IOC$DISMOUNT.
0000    35  ;
0000    36  ;       V03-014 HH0048          Hai Huang               15-Aug-1984
0000    37  ;               Deallocate the device on last channel deassign if the
0000    38  ;               device owner has gone away.
0000    39  ;
0000    40  ;       V03-013 ACG0441         Andrew C. Goldstein,    8-Aug-1984  15:16
0000    41  ;               Remove foreign dismount logic (moved to DISMOUNT)
0000    42  ;
0000    43  ;       V03-012 HH0033          Hai Huang               06-Jul-1984
0000    44  ;               Do foreign volume cleanups if the device is allocated
0000    45  ;               to the top level process in the process tree.
0000    46  ;
0000    47  ;       V03-011 HH0023          Hai Huang               05-Jun-1984
0000    48  ;               Correctly handle last channel deassign on an allocated
0000    49  ;               device.
0000    50  ;
0000    51  ;       V03-010 LMP0221         L. Mark Pilant,         30-Mar-1984  15:42
0000    52  ;               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
0000    53  ;               ORB$W_PROT.
0000    54  ;
0000    55  ;       V03-009 ACG0399         Andrew C. Goldstein,    24-Feb-1984  22:59
0000    56  ;               Incorporate I/O database locking rewrite, move LAST_CHAN
0000    57  ;               routine to IOSUBNPAG so it can be shared by DEALLOCATE.
```

```
0000     58 ;                  correct flow in deallocating device on dismount.
0000     59 ;
0000     60 ;       V03-008 CDS0001         Christian D. Saether    22-Sep-1983
0000     61 ;                Move deallocate on dismount action here from ioc$dismount
0000     62 ;                so that the device deallaction does not occur until
0000     63 ;                the last channel goes away.
0000     64 ;
0000     65 ;       V03-007 JLV0301         Jake VanNoy             30-JUL-1983
0000     66 ;                Add second call to IOC$VERIFYCHAN to prevent double
0000     67 ;                deassign.
0000     68 ;
0000     69 ;       V03-006 TCM0002         Trudy C. Matthews       28-Jun-1983
0000     70 ;                Small change to interface to EXE$UNLOCK_DEV.
0000     71 ;
0000     72 ;       V03-005 ROW0189         Ralph O. Weber          21-JUN-1983
0000     73 ;                Correct action taken upon discovery of a pending kernel mode
0000     74 ;                AST during "wait for all outstanding I/O to finish" logic.
0000     75 ;                Change method for lowering IPL to 0 from SETIPL to REI.  The
0000     76 ;                intent of lowering IPL is to allow the kernel mode AST to be
0000     77 ;                delivered.  However, the AST will be delivered only if IPL is
0000     78 ;                lowered via an REI.  (Only the REI instruction delivers AST
0000     79 ;                "interrupts.")
0000     80 ;
0000     81 ;       V03-004 TCM0001         Trudy C. Matthews       17-May-1983
0000     82 ;                Dequeue cluster-wide lock on last channel de-assign.
0000     83 ;
0000     84 ;       V03-003 ROW0170         Ralph O. Weber          12-MAR-1983
0000     85 ;                Reorder actual deassignment logic to be consistant with the
0000     86 ;                notion that the mailbox driver should perform all mailbox
0000     87 ;                dependent processing in its cancel I/O routine.  Setup use of
0000     88 ;                CAN$C_AMBXDGN cancel reason code to signal mailbox driver when
0000     89 ;                it is being called due to last reference deassignment on an
0000     90 ;                associated mailbox.
0000     91 ;
0000     92 ;       V03-002 ROW0127         Ralph O. Weber          5-OCT-1982
0000     93 ;                Make changes required to use new UCB creation and deletion
0000     94 ;                routines in UCBCREDEL.  Modify the delete-UCB check at the end
0000     95 ;                of this module to call IOC$DELMBX if the device independent
0000     96 ;                characteristics indicate a mailbox or call IOC$DELETE_UCB if
0000     97 ;                the device independent characteristics do not indicate a
0000     98 ;                mailbox.
0000     99 ;
0000    100 ;       V03-001 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    101 ;                Added $DEVDEF.
0000    102 ;
0000    103 ; SYSTEM SERVICE DEASSIGN I/O CHANNEL
0000    104 ;
0000    105 ; MACRO LIBRARY CALLS
0000    106 ;
0000    107
0000    108         $CANDEF                         ;DEFINE CANCEL REASON CODES
0000    109         $CCBDEF                         ;DEFINE CCB OFFSETS
0000    110         $DDBDEF                         ;DEFINE DDB OFFSETS
0000    111         $DDTDEF                         ;DEFINE DDT OFFSETS
0000    112         $DEVDEF                         ;DEFINE DEVICE TYPES
0000    113         $IODEF                          ;DEFINE I/O FUNCTION CODES
0000    114         $IPLDEF                         ;DEFINE INTERRUPT PRIORITY LEVELS
```

```
                     0000    115          $JIBDEF                          ;DEFINE JIB OFFSETS
                     0000    116          $ORBDEF                          ;DEFINE OBJECT'S RIGHTS BLOCK OFFSETS
                     0000    117          $PCBDEF                          ;DEFINE PCB OFFSETS
                     0000    118          $PRDEF                           ;DEFINE PROCESSOR REGISTERS
                     0000    119          $RSNDEF                          ;DEFINE RESOURCE WAIT NUMBERS
                     0000    120          $SSDEF                           ;DEFINE SYSTEM STATUS VALUES
                     0000    121          $UCBDEF                          ;DEFINE UCB OFFSETS
                     0000    122
                     0000    123 ;
                     0000    124 ; LOCAL SYMBOLS
                     0000    125 ;
                     0000    126 ; ARGUMENT LIST OFFSET DEFINITIONS
                     0000    127 ;
                     0000    128
           00000004  0000    129 CHAN=4                                    ;I/O CHANNEL NUMBER
                     0000    130
                     0000    131
                 00000000    132          .PSECT  AEXENONPAGED
```

```
                                                     .SBTTL  DEASSIGN I/O CHANNEL
                                                ;+
                                                ; EXE$DASSGN - DEASSIGN I/O CHANNEL
                                                ;
                                                ; THIS SERVICE DEASSIGNS A PREVIOUSLY ASSIGNED I/O CHANNEL AND CLEARS THE
                                                ; LINKAGE AND CONTROL INFORMATION IN THE CORRESPONDING CHANNEL CONTROL BLOCK.
                                                ; IF ANY I/O IS OUTSTANDING ON THE CHANNEL IT IS CANCELLED. IF A FILE IS
                                                ; OPEN ON THE CHANNEL IT IS CLOSED. IF A MAILBOX WAS ASSOCIATED WITH THE
                                                ; DEVICE WHEN IT WAS ASSIGNED, THE LINKAGE TO THE MAILBOX IS CLEARED. IF THE
                                                ; THE CHANNEL IS LAST ONE ASSIGNED TO THE DEVICE AND IT IS MARKED FOR DIS-
                                                ; MOUNT, THEN THE DISMOUNT IS COMPLETED.
                                                ;
                                                ; INPUTS:
                                                ;
                                                ;       CHAN(AP) = NUMBER OF THE I/O CHANNEL TO DEASSIGN.
                                                ;
                                                ;       R4 = CURRENT PROCESS PCB ADDRESS.
                                                ;
                                                ; OUTPUTS:
                                                ;
                                                ;       R0 LOW BIT CLEAR INDICATES FAILURE TO DEASSIGN CHANNEL.
                                                ;
                                                ;               R0 = SS$_IVCHAN - INVALID CHANNEL NUMBER SPECIFIED.
                                                ;
                                                ;               R0 = SS$_NOPRIV - SPECIFIED CHANNEL IS NOT ASSIGNED TO A
                                                ;                       DEVICE OR THE CALLER DOES NOT HAVE SUFFICIENT
                                                ;                       PRIVILEGE TO ACCESS THE CHANNEL.
                                                ;
                                                ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                                                ;
                                                ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                                                ;-
                                                ;
                                01FC            .ENTRY  EXE$DASSGN,^M<R2,R3,R4,R5,R6,R7,R8>
                                55   04 AC     MOVZWL  CHAN(AP),R5             ;GET CHANNEL NUMBER
                                50   55   D0    MOVL    R5,R0                   ;COPY I/O CHANNEL NUMBER
                                   FFF4' 30     BSBW    IOC$VERIFYCHAN          ;VERIFY CHANNEL NUMBER
                                  63 50  E9     BLBC    R0,50$                  ;IF LBC INVALID CHANNEL
                                56   51   D0    MOVL    R1,R6                   ;COPY ADDRESS OF CCB
                                57   52   D0    MOVL    R2,R7                   ;SAVE CHANNEL INDEX
                                7E   01   9A    MOVZBL  #CAN$C_DASSGN,-(SP)     ;PUSH DEASSIGN CODE
                                7E   55   3C    MOVZWL  R5,-(SP)                ;PUSH CHANNEL NUMBER
                00000000'EF     02   FB         CALLS   #2,EXE$CANCELN          ;CANCEL I/O ON CHANNEL
                                                ;
                                                ; Channel is verified again. This is because the $CANCEL could
                                                ; have activated a kernel mode AST routine which did another
                                                ; $DASSGN.  (This can happen in $BRKTHRU, for example)
                                                ;
                                50   55   D0    MOVL    R5,R0                   ;COPY I/O CHANNEL NUMBER
                                   FFD8' 30     BSBW    IOC$VERIFYCHAN          ;VERIFY CHANNEL NUMBER
                                  47 50  E9     BLBC    R0,50$                  ;IF LBC INVALID CHANNEL
                                  04 A6  D5     TSTL    CCB$L_WIND(R6)          ;FILE ACCESSED ON CHANNEL?
                                     25  13     BEQL    30$                     ;IF EQL NO
                                                $QIOW_S #30,R5,#IO$_DEACCESS    ;DEACCESS FILE
                                  07 50  E8     BLBS    R0,30$                  ;IF LBS SUCCESSFUL COMPLETION
                             50   00AC 8F  B1    CMPW    #SS$_FILNOTACC,R0       ;NO FILE ACCESSED?
                                     1D  12     BNEQ    50$                     ;IF NEQ NO
```

```
                                                E 7
                7E   DC  0055   191 30$:    MOVPSL  -(SP,                   ;SAVE CURRENT PROCESSOR STATUS
                         0057   192         SETIPL  #IPL$_ASTDEL            ;RAISE TO AST DELIVERY LEVEL
              0A A6   B5  005A   193         TSTW    CCB$W_IOC(R6)           ;ANY I/O STILL OUTSTANDING?
                 18   13  005D   194         BEQL    60$                     ;IF EQL NO
                         005F   195         SETIPL  #IPL$_SYNCH             ;RAISE TO SYNCHRONIZATION LEVEL
           50    13   DB  0062   196         MFPR    #PR$_ASTLVL,R0          ;READ CURRENT AST LEVEL
                 0C   13  0065   197         BEQL    55$                     ;IF EQL KERNEL AST QUEUED
           50    01   3C  0067   198         MOVZWL  #RSN$_ASTWAIT,R0        ;SET AST WAIT RESOURCE WAIT NUMBER
              FF93'   30  006A   199         BSBW    SCH$RQWAIT              ;WAIT FOR AST
                         006D   200 40$:     SETIPL  #0                      ;ALLOW INTERRUPTS
                 B9   11  0070   201         BRB     20$                     ;
                      04  0072   202 50$:     RET                             ;
                         0073   203
              B5 AF   9F  0073   204 55$:     PUSHAB  B^20$                   ;ALLOW KERNEL AST TO BE DELIVERED
                      02  0076   205         REI                             ;AND CONTINUE AT 20$
                         0077   206
                         0077   207 ;
                         0077   208 ; DEASSIGN CHANNEL
                         0077   209 ;
              FF86'   30  0077   210 60$:     BSBW    SCH$IOLOCKW             ;LOCK I/O DATABASE FOR WRITE ACCESS
           55    66   D0  007A   211         MOVL    CCB$L_UCB(R6),R5        ;GET ASSIGNED DEVICE UCB ADDRESS
              09 A6   94  007D   212         CLRB    CCB$B_AMOD(R6)          ;DEASSIGN CHANNEL
                         0080   213 ;
                         0080   214 ; CHECK IF CHANNEL ASSOCIATED A MAILBOX AND IF SO, CLEAN IT UP
                         0080   215 ;
        19 08 A6    00   E1  0080   216         BBC     #CCB$V_AMB, -           ; Branch if no MBX associated
                         0085   217                 CCB$B_STS(R6), 70$      ; by this channel.
           53    55   D0  0085   218         MOVL    R5, R3                  ; Copy address of device UCB.
              55    60 A5   D0  0088   219         MOVL    UCB$L_AMB(R5), R5       ; Get associated mailbox UCB address.
                 10   13  008C   220         BEQL    70$                     ; Branch if none.
        0B 38 A5    14   E1  008E   221         BBC     #DEV$V_MBX, -           ; Branch if associated device is
                         0093   222                 UCB$L_DEVCHAR(R5), 70$  ; not a mailbox.
              60 A3   D4  0093   223         CLRL    UCB$L_AMB(R3)           ; Clear associated mailbox addresss.
              5C A5   B7  0096   224         DECW    UCB$W_REFC(R5)          ; Decrement mailbox UCB reference count.
                 03   12  0099   225         BNEQ    70$                     ; Branch if mailbox still referenced.
                FF62'   30  009B   226         BSBW    IOC$LAST_CHAN_AMBX      ; Else, process last channel deassign
                         009E   227                                         ; for the associated mailbox.
                         009E   228
                         009E   229 ;
                         009E   230 ; DECREMENT REFERENCE COUNT
                         009E   231 ; CHECK FOR AND PROCESS LAST CHANNEL DEASSIGN
                         009E   232 ;
              55    66   D0  009E   233 70$:     MOVL    CCB$L_UCB(R6), R5       ; Get device UCB address back.
                 5C A5   B7  00A1   234         DECW    UCB$W_REFC(R5)          ; Decrement device reference count.
                    0D   12  00A4   235         BNEQ    80$                     ; Branch if not very last reference.
                         00A6   236 ;
                         00A6   237 ; Handle ref count reaching zero.
                         00A6   238 ;
                 2C A5   D4  00A6   239         CLRL    UCB$L_PID(R5)           ; Clear owner process id.
        10 3C A5    00   E1  00A9   240         BBC     #DEV$V_CLU, -           ; Branch if device not available
                         00AE   241                 UCB$L_DEVCHAR2(R5),100$ ; cluster-wide.
                FF4F'   30  00AE   242         BSBW    IOC$UNLOCK_DEV          ; Dequeue cluster-wide device lock.
                   0B   11  00B1   243         BRB     100$                    ; Do rest of last channel deassign
                         00B3   244 ;
                         00B3   245 ; Check for last channel deassign on an allocated device
                         00B3   246 ;
           5C A5    01   B1  00B3   247 80$:     CMPW    #1, UCB$W_REFC(R5)      ; UCB reference count one?
```

```
                 0B   12  00B7   248          BNEQ    110$                    ; If not one, branch to finish request
                                  00B9   249
          06 38 A5   17  E1  00B9   250          BBC     #DEV$V_ALL, -           ; If device not allocated,
                                  00BE   251                  UCB$L_DEVCHAR(R5), 110$ ; branch to finish request.
                                  00BE   252 ;
                                  00BF   253 ; Call driver's cancel I/O routine with CAN$C_DASSGN reason code and
                                  00BE   254 ; if appropriate, delete UCB
                                  00BE   255 ;
                52   57  D0  00BE   256 100$:   MOVL    R7,R2                   ; Get channel index
                   FF3C'  30  00C1   257          BSBW    IOC$LAST_CHAN           ; Do common last channel deassign.
                                  00C4   258 ;
                                  00C4   259 ; SET STATUS AND EXIT
                                  00C4   260 ;
                50   01  3C  00C4   261 110$:   MOVZWL  #SS$_NORMAL, R0         ; Set normal completion.
                   FF36'  31  00C7   262          BRW     IOC$UNLOCK              ; Unlock I/O data base and return.
                                  00CA   263
                                  00CA   264
                                  00CA   265          .END
```

```
$$T1                            = 00000001
CAN$C_DASSGN                    = 00000001
CCB$B_AMOD                      = 00000009
CCB$B_STS                       = 00000008
CCB$L_UCB                       = 00000000
CCB$L_WIND                      = 00000004
CCB$V_AMB                       = 00000000
CCB$W_IOC                       = 0000000A
CHAN                            = 00000004
DEV$V_ALL                       = 00000017
DEV$V_CLU                       = 00000000
DEV$V_MBX                       = 00000014
EXE$CANCELN                       ********    X   02
EXE$DASSGN                        00000000 RG     02
IO$_DEACCESS                    = 00000034
IOC$LAST_CHAN                     ********    X   02
IOC$LAST_CHAN_AMBX                ********    X   02
IOC$UNLOCK                        ********    X   02
IOC$UNLOCK_DEV                    ********    X   02
IOC$VERIFYCHAN                    ********    X   02
IPL$_ASTDEL                     = 00000002
IPL$_SYNCH                      = 00000008
PR$_ASTLVL                      = 00000013
PR$_IPL                         = 00000012
RSN$_ASTWAIT                    = 00000001
SCH$IOLOCKW                       ********    X   02
SCH$RWAIT                         ********    X   02
SS$_FILNOTACC                   = 000000AC
SS$_NORMAL                      = 00000001
SYS$QIOW                          ********   GX   02
UCB$L_AMB                       = 00000060
UCB$L_DEVCHAR                   = 00000038
UCB$L_DEVCHAR2                  = 0000003C
UCB$L_PID                       = 0000002C
UCB$W_REFC                      = 0000005C
```

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	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	000000CA (202.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.04	00:00:01.77
Command processing	110	00:00:00.49	00:00:05.71
Pass 1	391	00:00:13.84	00:00:40.77
Symbol table sort	6	00:00:02.46	00:00:09.56
Pass 2	67	00:00:02.29	00:00:05.78

Symbol table output	6	00:00:00.07	00:00:00.07
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	613	00:00:19.21	00:01:03.69

The working set limit was 1500 pages.
79726 bytes (156 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1600 non-local and 10 local symbols.
265 source lines were read in Pass 1, producing 16 object records in Pass 2.
26 pages of virtual memory were used to define 25 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	11
_255DUA28:[SYSLIB]STARLET.MLB;2	11
TOTALS (all libraries)	22

1758 GETS were required to define 22 macros.

There were no errors, warnings or information messages.

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

SYSCRMPSC
LIS

SYSDCLEXH
LIS

SYSDEVALC
LIS

SYSCVTIIM
LIS

SYSDGBLSC
LIS

SYSENQDEQ
LIS

SYSDCLCMH
LIS

SYSDERLMB
LIS

SYSDASSGN
LIS

SYSDELPRC
LIS