```
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDD          DDD  CCC            LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
```

FILEID**SYSOUTPUT

```
  SSSSSSSS  YY      YY   SSSSSSSS    000000    UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT
  SSSSSSSS  YY      YY   SSSSSSSS    000000    UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT
SS          YY      YY  SS          00    00  UU      UU      TT      PP      PP  UU      UU      TT
SS          YY      YY  SS          00    00  UU      UU      TT      PP      PP  UU      UU      TT
SS            YY  YY    SS          00    00  UU      UU      TT      PP      PP  UU      UU      TT
SS            YY  YY    SS          00    00  UU      UU      TT      PP      PP  UU      UU      TT
  SSSSSS        YY        SSSSSS    00    00  UU      UU      TT      PPPPPPPP    UU      UU      TT
  SSSSSS        YY        SSSSSS    00    00  UU      UU      TT      PPPPPPPP    UU      UU      TT
        SS      YY              SS  00    00  UU      UU      TT      PP          UU      UU      TT
        SS      YY              SS  00    00  UU      UU      TT      PP          UU      UU      TT
        SS      YY              SS  00    00  UU      UU      TT      PP          UU      UU      TT      ....
        SS      YY              SS  00    00  UU      UU      TT      PP          UU      UU      TT      ....
SSSSSSSS        YY      SSSSSSSS    000000    UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT      ....
SSSSSSSS        YY      SSSSSSSS    000000    UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT      ....
```

```
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
```

Table of contents

```
                                .TITLE  SYSOUTPUT - ROUTINES FOR MANIPULATING SYS$OUTPUT
                                .IDENT  'V04-000'

                        ;
                        ;*****************************************************************************
                        ;*                                                                           *
                        ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
                        ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                        ;*  ALL RIGHTS RESERVED.                                                     *
                        ;*                                                                           *
                        ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
                        ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
                        ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
                        ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
                        ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
                        ;*  TRANSFERRED.                                                             *
                        ;*                                                                           *
                        ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
                        ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
                        ;*  CORPORATION.                                                             *
                        ;*                                                                           *
                        ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
                        ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
                        ;*                                                                           *
                        ;*****************************************************************************
                        ;
                        ; Peter George  08-Jan-1983
                        ;
                        ; ROUTINES FOR MANIPULATING SYS$OUTPUT
                        ;
                        ; MODIFIED BY:
                        ;
                        ;       V03-004 PCG0004         Peter George            21-Aug-1983
                        ;               Manage concealed and terminal attributes using new
                        ;               logical name services.
                        ;
                        ;       V03-003 PCG0003         Peter George            27-Jul-1983
                        ;               Fix data manipulation bugs.
                        ;
                        ;       V03-002 PCG0002         Peter George            27-May-1983
                        ;               Add support for image verification.
                        ;
                        ;       V03-001 PCG0001         Peter George            13-Apr-1983
                        ;               Clear FAB$B_SHR before opening SYS$OUTPUT.
                        ;---
                        ;
                        ; MACRO LIBRARY CALLS
                        ;

                                PRCDEF                          ;DEFINE PROCESS WORK AREA
                                IDFDEF                          ;DEFINE INDIRECT FRAME OFFSETS
                                PRDDEF                          ;PROCESS RMS DATA
                                OUTDEF                          ;SYS$OUTPUT DEFINITIONS
                                $CLIMSGDEF                      ;DEFINE ERROR/STATUS VALUES
                                $FABDEF                         ;DEFINE FAB OFFSETS
```

```
                                0000     58          $RABDEF                         ;DEFINE RAB OFFSETS
                                0000     59          $LNMDEF                         ;DEFINE LNM OFFSET
                                0000     60          $NAMDEF                         ;DEFINE NAM OFFSETS
                                0000     61          $PSLDEF                         ;DEFINE PSL FIELDS
                                0000     62
                                0000     63 ;
                                0000     64 ; LOCAL DATA
                                0000     65 ;
                                0000     66
                            00000000     67                  .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
                                0000     68 INPUTNAM:
      54 55 50 4E 49 24 53 59 53 09     0000     69                  .ASCII  <9>/SYS$INPUT/          ;'INPUT' LOGICAL NAME TEXT
                                000A     70 OUTFILE:                                             ;
                        53 49 4C 2E     000A     71                  .ASCII  /.LIS/                  ;OUTPUT FILE DEFAULT NAME STRING
                                000E     72 OUTPUTNAM:                                           ;
   54 55 50 54 55 4F 24 53 59 53 0A     000E     73                  .ASCII  <10>/SYS$OUTPUT/        ;'OUTPUT' LOGICAL NAME TEXT
                                0019     74 LNM$PROCESS:                                         ;
53 53 45 43 4F 52 50 24 4D 4E 4C 00'    0019     75                  .ASCIC  /LNM$PROCESS/           ;PROCESS LOGICAL NAME TABLE
                                0B      0019                                                     ;
```

```
                     0025    77          .SBTTL  OPEN NEW SYS$OUTPUT FILE
                     0025    78 ;+
                     0025    79 ; DCL$OPEN_OUTPUT - OPEN NEW SYS$OUTPUT FILE
                     0025    80 ;
                     0025    81 ; THIS ROUTINE IS CALLED TO OPEN A NEW SYS$OUTPUT FILE AND TO INITIALIZE
                     0025    82 ; THE APPROPRIATE DATA STRUCTURES.
                     0025    83 ;
                     0025    84 ; INPUTS:
                     0025    85 ;
                     0025    86 ;       R1/R2 = DESCRIPTOR OF SYS$OUTPUT EQUIVALENCE NAME
                     0025    87 ;       R6 = ADDRESS OF DEFAULT IDF FRAME
                     0025    88 ;       R11 = ADDRESS OF PROCESS WORK AREA
                     0025    89 ;
                     0025    90 ; OUTPUTS:
                     0025    91 ;
                     0025    92 ;       R0 = STATUS (NOT SIGNALED)
                     0025    93 ;-
                     0025    94
                     0025    95 DCL$OPEN_OUTPUT::
          03C6 8F    BB  0025    96          PUSHR   #^M<R1,R2,R6,R7,R8,R9>
                     0029    97
     58   00BC CB    D0  0029    98          MOVL    PRC_L_IDFLNK(R11),R8            ;GET ADDRESS OF CURRENT IDF FRAME
               51    D5  002E    99          TSTL    R1                              ;OUTPUT FILE SPECIFIED?
               20    12  0030   100          BNEQ    10$                             ;BRANCH IF SO
                     0032   101
                     0032   102 ;
                     0032   103 ; No output filespec is specified, propagate the default output stream
                     0032   104 ; to the indirect frame.  Leave PRC_L_INDOUTRAB as is.
                     0032   105 ;
                     0032   106          ASSUME  IDF_W_OUTISI EQ IDF_W_OUTIFI+2
            20 A6    D0  0032   107          MOVL    IDF_W_OUTIFI(R6),-              ;SAVE PREVIOUS INTERNAL FILE INDEX
            20 A8        0035   108                  IDF_W_OUTIFI(R8)                ;
            24 A6    D0  0037   109          MOVL    IDF_L_OUTRABCTX(R6),-           ;SAME FOR RAB CONTEXT
            24 A8        003A   110                  IDF_L_OUTRABCTX(R8)             ;
        28 A6    10  28  003C   111          MOVC    #16,IDF_T_OUTDVI(R6),-          ;SAVE PREVIOUS DEVICE NAME
            28 A8        0040   112                  IDF_T_OUTDVI(R8)                ;
                     0042   113          CLRBIT  IDF_V_OUTCCL,IDF_B_OUTFLAGS(R8) ;CLEAR CONCEALED BIT
               0C    E1  0046   114          BBC     #NAM$V_CNCL_DEV,-               ;IS DEVICE CONCEALED?
         04 34 A7        0048   115                  NAM$L_FNB(R7),5$                ;
                     004B   116          SETBIT  IDF_V_OUTCCL,IDF_B_OUTFLAGS(R8) ;SET CONCEALED BIT IN IDF
            0088     31  004F   117 5$:     BRW     30$                             ;SKIP PAST FILE CREATION
                     0052   118
                     0052   119 ;
                     0052   120 ; Output filespec is specified, create new output file.  fill in IDF frame
                     0052   121 ; fields and PRC_L_INDOUTRAB.
                     0052   122 ;
                     0052   123 ; Set up fields in the indirect FAB for the $CREATE.
                     0052   124 ;
     59    1C AB    D0  0052   125 10$:     MOVL    PRC_L_INDFAB(R11),R9            ;GET ADDRESS OF INDIRECT FAB
 30 A9     B1 AF    9E  0056   126          MOVAB   OUTFILE,FAB$L_DNA(R9)           ;SET ADDRESS OF DEF NAME STRING (.LI
   16 A9     01    90  005B   127          MOVB    #FAB$M_PUT,FAB$B_FAC(R9)        ;SET FILE ACCESS TYPE
   34 A9     51    90  005F   128          MOVB    R1,FAB$B_FNS(R9)                ;SET SIZE OF FILENAME STRING
   2C A9     52    D0  0063   129          MOVL    R2,FAB$L_FNA(R9)                ;SET ADDRESS OF FILE NAME STRING
 00040040    8F    D0  0067   130          MOVL    #FAB$M_PPF!FAB$M_SQO,-          ;SET FILE OPEN OPTIONS
         04 A9        006D   131                  FAB$L_FOP(R9)                   ;
   1D A9     00    90  006F   132          MOVB    #FAB$C_SEQ,FAB$B_ORG(R9)        ;SET FILE ORGANIZATION TYPE
   1E A9     04    90  0073   133          MOVB    #FAB$M_PRN,FAB$B_RAT(R9)        ;SET RECORD ATTRIBUTE TYPE TO PRINT
```

```
                1F A9    03    90   0077    134          MOVB    #FAB$C_VFC,FAB$B_RFM(R9)          ;SET RECORD FORMAT TYPE-FIXED CONTRO
                   17 A9      94   007B    135          CLRB    FAB$B_SHR(R9)                     ;CLEAR FILE SHARING OPTIONS
                   02 A9      B4   007E    136          CLRW    FAB$W_IFI(R9)                     ;CLEAR INTERNAL FILE INDEX
                50    59      D0   0081    137          MOVL    R9,R0                             ;ADDRESS OF FAB
                51    01      D0   0084    138          MOVL    #1,R1                             ;ASSUME CREATE WITH ERROR REPORTING
                      FF76'   30   0087    139          BSBW    DCL$OPEN_CREATE                   ;CREATE NEW OUTPUT FILE
                   65 50      E9   008A    140          BLBC    R0,50$                            ;IF LBC CREATION FAILURE
                                   008D    141
                                   008D    142 ;
                                   008D    143 ; The $CREATE was successful, now set up the RAB fields and try a $CONNECT.
                                   008D    144 ; be careful to restore IDF_W_OUTIFI and IDF_L_OUTRABCTX if anything goes wrong.
                                   008D    145 ;
             52    0138 C9    9E   008D    146          MOVAB   PRD_G_ALTOUTRAB(R9),R2            ;GET ALTERNATE OUTPUT RAB ADDRESS
                   02 A2      B4   0092    147          CLRW    RAB$W_ISI(R2)                     ;CLEAR INTERNAL STREAM INDEX
                3C A2    59    D0   0095    148          MOVL    R9,RAB$L_FAB(R2)                  ;LINK FAB TO RAB
                                   0099    149          $CONNECT RAB=(R2)                         ;CONNECT TO OUTPUT
                   4D 50      E9   00A2    150          BLBC    R0,50$                            ;IF LBS CONNECT FAILURE
                                   00A5    151
                                   00A5    152 ;
                                   00A5    153 ; Both the $CREATE and $CONNECT succeeded.  Fill in the IDF and RAB fields.
                                   00A5    154 ; Set PRC_L_INDOUTRAB.
                                   00A5    155 ;
                18 AB    62    9E   00A5    156          MOVAB   (R2),PRC_L_INDOUTRAB(R11)         ;SET ADDRESS OF OUTPUT RAB
             20 A8    02 A9    B0   00A9    157          MOVW    FAB$W_IFI(R9),IDF_W_OUTIFI(R8)    ;SAVE OUTPUT FILE INTERNAL INDEX
             18 A2    40 A9    D0   00AE    158          MOVL    FAB$L_DEV(R9),RAB$L_CTX(R2)       ;SAVE DEVICE CHARACTERISTICS
                      18 A2    D0   00B3    159          MOVL    RAB$L_CTX(R2),-                   ;  IN RAB AND IDF
                      24 A8             00B6    160                  IDF_L_OUTRABCTX(R8)          ;
                   06    02    F0   00B8    161          INSV    #FAB$M_CR,#RAB$V_PPF_RAT,-        ;SET TO USE IMPLIED
                02 A2       08        00BB    162                  #RAB$S_PPF_RAT,RAB$W_ISI(R2)      ;  CR/LF FOR OUTPUT
             22 A8    02 A2    B0   00BE    163          MOVW    RAB$W_ISI(R2),IDF_W_OUTISI(R8)    ;COPY OUTPUT ISI TO IDF
                                   00C3    164
                                   00C3    165 ;
                                   00C3    166 ; Get the device name.  Propagate concealed attribute.
                                   00C3    167 ;
                57    28 A9      D0   00C3    168          MOVL    FAB$L_NAM(R9),R7                  ;GET ADDRESS OF INDIRECT NAM
                                   00C7    169          CLRBIT  IDF_V_OUTCCL,IDF_B_OUTFLAGS(R8)   ;CLEAR CONCEALED BIT IN IDF
                         0C    E1   00CB    170          BBC     #NAM$V_CNCL_DEV,-                 ;IS DEVICE CONCEALED?
                   04 34 A7          00CD    171                  NAM$L_FNB(R7),20$                 ;
                                   00D0    172          SETBIT  IDF_V_OUTCCL,IDF_B_OUTFLAGS(R8)   ;SET CONCEALED BIT IN IDF
             14 A7    10    28   00D4    173 20$:     MOVC    #16,NAM$T_DVI(R7),-               ;SAVE DEVICE NAME
                   28 A8          00D8    174                  IDF_T_OUTDVI(R8)                  ;
                                   00DA    175
                                   00DA    176 ;
                                   00DA    177 ; Redirect image verification if appropriate.
                                   00DA    178 ;
                51    1C AB      D0   00DA    179 30$:     MOVL    PRC_L_INDFAB(R11),R1              ;GET ADDRESS OF INDIRECT FAB
             02 A1    04 A8      B0   00DE    180          MOVW    IDF_W_INPIFI(R8),FAB$W_IFI(R1)    ;GET INPUT IFI
                      FF1A'      30   00E3    181          BSBW    DCL$VERIFY_IMAGE                  ;
                                   00E6    182
                                   00E6    183 ;
                                   00E6    184 ; Cleanup and exit.
                                   00E6    185 ;
                                   00E6    186          STATUS  NORMAL                            ;
                   03C6 8F    BA   00ED    187 40$:     POPR    #^M<R1,R2,R6,R7,R8,R9>            ;RESTORE REGISTERS
                              05   00F1    188          RSB
                                   00F2    189
                                   00F2    190 ;
```

```
                                                C 7
                     00F2   191 ; Restore corrupted FAB and RAB fields.
                     00F2   192 ;
   59   1C AB   D0  00F2   193 50$:    MOVL    PRC_L_INDFAB(R11),R9                    ;GET ADDRESS OF INDIRECT FAB
02 A9   20 A8   B0  00F6   194          MOVW    IDF_W_OUTIFI(R8),FAB$W_IFI(R9)          ;CLEAR INTERNAL FILE INDEX
 52  0138 C9   9E  00FB   195          MOVAB   PRD_G_ALTOUTRAB(R9),R2                  ;GET ALTERNATE OUTPUT RAB ADDRESS
02 A2   22 A8   B0  0100   196          MOVW    IDF_W_OUTISI(R8),RAB$W_ISI(R2)          ;CLEAR INTERNAL STREAM INDEX
         E6   11  0105   197          BRB     40$                                     ;
```

```
                                0107   199          .SBTTL  RESTORE OLD SYS$OUTPUT FILE
                                0107   200 ;+
                                0107   201 ; DCL$RESTORE_OUTPUT - RESTORE OLD SYS$OUTPUT FILE
                                0107   202 ;
                                0107   203 ; THIS ROUTINE IS CALLED TO CLOSE THE CURRENT SYS$OUTPUT FILE AND TO RESET
                                0107   204 ; THE APPROPRIATE DATA STRUCTURES SO THAT THEY POINT TO THE OLD SYS$OUTPUT
                                0107   205 ; FILE.
                                0107   206 ;
                                0107   207 ; INPUTS:
                                0107   208 ;
                                0107   209 ;       R2 = ADDRESS OF OUTPUT FILE DATA
                                0107   210 ;       R8 = ADDRESS OF CURRENT IDF FRAME
                                0107   211 ;       R11 = ADDRESS OF PROCESS WORK AREA
                                0107   212 ;
                                0107   213 ; OUTPUTS:
                                0107   214 ;
                                0107   215 ;       R0-R1 ARE TRASHED
                                0107   216 ;-
                                0107   217
                                0107   218 DCL$RESTORE_OUTPUT::                          ;RESTORE SYS$OUTPUT
                          59    DD    0107   219          PUSHL   R9                      ;SAVE R9
                    59    1C AB    D0    0109   220          MOVL    PRC_L_INDFAB(R11),R9    ;GET ADDRESS OF SCRATCH FAB
                                0109   221
                                0109   222 ;
                                0109   223 ; Close current output file if the current output file is different from
                                0109   224 ; the previous level and different from the process permanent output file.
                                0109   225 ;
                          68    D5    0109   226          TSTL    IDF_L_LNK(R8)           ;DOES PREVIOUS LEVEL EXIST?
                          08    13    010B   227          BEQL    5$                      ;NO, THEN SKIP NEXT TEST
                       20 A8    B1    010D   228          CMPW    IDF_W_OUTIFI(R8),-      ;OUTPUT FILE CHANGED FROM
                    0094 C8          0110   229                  IDF_W_OUTIFI+IDF_K_LENGTH(R8)   ;  PREVIOUS COMMAND LEVEL?
                          16    13    0113   230          BEQL    10$                     ;BR IF NO
                       20 A8    B1    0115   231 5$:      CMPW    IDF_W_OUTIFI(R8),-      ;OUTPUT FILE DIFFERENT FROM
                    0114 CB          0118   232                  PRC_W_OUTIFI(R11)       ;  PERMANENT PROCESS DEFINITION?
                          0E    13    011B   233          BEQL    10$                     ;BR IF NO
                02 A9  20 A8    B0    011D   234          MOVW    IDF_W_OUTIFI(R8),FAB$W_IFI(R9)  ;RESTORE INTERNAL FILE INDEX
                                0122   235          $CLOSE  FAB=(R9)                    ;CLOSE INDIRECT OUTPUT FILE
                                012B   236
                                012B   237 ;
                                012B   238 ; Restore the IDF and the RAB.  Use either PRC or some other IDF, as specified
                                012B   239 ; by the address in R2.
                                012B   240 ;
                    50    18 AB    D0    012B   241 10$:      MOVL    PRC_L_INDOUTRAB(R11),R0 ;GET POINTER TO INDIRECT OUTPUT RAB
                02 A0    02 A2    B0    012F   242          MOVW    OUT_W_OUTISI(R2),RAB$W_ISI(R0)  ;SAVE ISI
                         04 A2    D0    0134   243          MOVL    OUT_L_OUTRABCTX(R2),-   ;SAVE CTX
                         18 A0          0137   244                  RAB$L_CTX(R0)           ;
                                0139   245          ASSUME  OUT_W_OUTISI EQ OUT_W_OUTIFI+2
                                0139   246          ASSUME  OUT_L_OUTRABCTX EQ OUT_W_OUTIFI+4
                    20 A8    62    7D    0139   247          MOVQ    OUT_W_OUTIFI(R2),IDF_W_OUTIFI(R8) ;SAVE IFI,ISI,RABCTX
                             3F    BB    013D   248          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ;SAVE VOLATILE REGISTERS
                08 A2        10    28    0141   249          MOVC    #16,OUT_T_OUTDVI(R2),-  ;SAVE DEVICE NAME
                         28 A8          0145   250                  IDF_T_OUTDVI(R8)        ;
                             3F    BA    0147   251          POPR    #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE VOLATILE REGISTERS
                                0149   252          CLRBIT  IDF_V_OUTCCL,IDF_B_OUTFLAGS(R8) ;CLEAR CONCEALED BIT IN IDF
                             00    E1    014D   253          BBC     #OUT_V_OUTCCL,-         ;IS DEVICE CONCEALED?
                      04 18 A2          0150   254                  OUT_B_OUTFLAGS(R2),20$  ;
                                0153   255          SETBIT  IDF_V_OUTCCL,IDF_B_OUTFLAGS(R8) ;SET CONCEALED BIT IN IDF
```

```
                                                E 7
                     0158   256
                     0158   257 ;
                     0158   258 ; Redirect image verification if appropriate.
                     0158   259 ;
      51   59   D0   0158   260 20$:    MOVL    R9,R1                                   ;GET ADDRESS OF FAB
02 A1   04 A8   B0   015B   261         MOVW    IDF_W_INPIFI(R8),FAB$W_IFI(R1)          ;GET INPUT IFI
        FE9D'   30   015E   262         BSBW    DCL$VERIFY_IMAGE                        ;
                     0161   263
           59 8ED0   0161   264         POPL    R9                                      ;RESTORE R9
                05   0164   265         RSB                                             ;
                     0165   266
```

```
                     0167   268          .SBTTL  CREATE LOGICAL NAMES FOR 'INPUT' AND 'OUTPUT'
                     0167   269 ;---
                     0167   270 ;
                     0167   271 ; DCL$CREATE_IO - CREATE LOGICAL NAMES FOR 'INPUT' AND 'OUTPUT'
                     0167   272 ; DCL$CREATE_OUTPUT - CREATE LOGICAL NAME FOR 'OUTPUT'
                     0167   273 ;
                     0167   274 ; THESE ROUTINES ARE CALLED TO CREATE LOGICAL NAMES FOR 'INPUT' AND 'OUTPUT'
                     0167   275 ; BASE ON THE CURRENT INDIRECT FILE DEPTH.
                     0167   276 ;
                     0167   277 ; INPUTS:
                     0167   278 ;
                     0167   279 ;       R8 = ADDRESS OF CURRENT INDIRECT FRAME
                     0167   280 ;       R11 = ADDRESS OF PROCESS WORK AREA
                     0167   281 ;
                     0167   282 ; OUTPUTS:
                     0167   283 ;
                     0167   284 ;       R0-R6 DESTROYED.
                     0167   285 ;---
                     0167   286
                     0167   287 DCL$CREATE_IO::
          52   3C A8  9E  0167   288          MOVAB   IDF_T_INPDVI(R8),R2             ;GET ADDRESS OF INPUT DEVICE IDENTIF
            4000 8F   AB  016B   289          BICW3   #FAB$M_PPF_IND,-                ;GET INTERNAL FILE ID FOR INPUT -
          53   04 A8      016F   290                  IDF_W_INPIFI(R8),R3             ;ALWAYS CLEAR INDIRECT BIT IF SET
     55   FE8A CF   9E  0172   291          MOVAB   INPUTNAM,R5                     ;GET ADDRESS OF INPUT LOGICAL NAME
          5C AB   D5  0177   292          TSTL    PRC_L_INDEPTH(R11)              ;ARE WE AT LEVEL 0?
            05    12  017A   293          BNEQ    10$                             ;NO, THEN SKIP
           009B   30  017C   294          BSBW    DELETE                          ;DELETE 'INPUT' LOGICAL NAME
             12   11  017F   295          BRB     DCL$CREATE_OUTPUT               ;CREATE 'OUTPUT' LOGICAL NAME
                     0181   296
56   00000200 8F   D0  0181   297 10$:     MOVL    #LNM$M_TERMINAL,R6              ;ASSUME NOT CONCEALED DEVICE
             01   E1  0188   298          BBC     #IDF_V_INPCCL,-                 ;SKIP IF NOT CONCEALED DEVICE
          04 38 A8      018A   299                  IDF_B_OUTFLAGS(R8),20$          ;
                     018D   300          SETBIT  LNM$V_CONCEALED,R6              ;SET ATTRIBUTES
             20   10  0191   301 20$:     BSBB    CREATE                          ;CREATE 'INPUT' LOGICAL NAME
                     0193   302
                     0193   303 DCL$CREATE_OUTPUT::
          52   28 A8  9E  0193   304          MOVAB   IDF_T_OUTDVI(R8),R2             ;GET ADDRESS OF OUTPUT DEVICE IDENTI
            4000 8F   AB  0197   305          BICW3   #FAB$M_PPF_IND,-                ;GET INTERNAL FILE ID FOR OUTPUT FIL
          53   20 A8      019B   306                  IDF_W_OUTIFI(R8),R3             ;ALWAYS CLEAR INDIRECT BIT IF SET
     55   FE6C CF   9E  019E   307          MOVAB   OUTPUTNAM,R5                    ;GET ADDRESS OF OUTPUT LOGICAL NAME
56   00000200 8F   D0  01A3   308          MOVL    #LNM$M_TERMINAL,R6              ;ASSUME NOT CONCEALED DEVICE
             00   E1  01AA   309          BBC     #IDF_V_OUTCCL,-                 ;SKIP IF NOT CONCEALED DEVICE
          04 38 A8      01AC   310                  IDF_B_OUTFLAGS(R8),CREATE       ;
                     01AF   311          SETBIT  LNM$V_CONCEALED,R6              ;SET ATTRIBUTES
                     01B3   312
                     01B3   313 ;
                     01B3   314 ; R2 = ADDRESS OF ASCIC DEVICE NAME
                     01B3   315 ; R3 = IFI CORRESPONDING TO FILE
                     01B3   316 ; R5 = ADDRESS OF ASCIC LOGICAL NAME
                     01B3   317 ; R6 = EQUIVALENCE NAME ATTRIBUTES
                     01B3   318 ;
                     01B3   319 ; CREATE EQUIVALENCE NAME STRING ON THE STACK.
                     01B3   320 ; FORMAT IS <ESC-BYTE><00-BYTE><IFI-WORD><DEVICE-STRING><:-BYTE>.
                     01B3   321 ;
          5E   10   C2  01B3   322 CREATE: SUBL    #16,SP                          ;ALLOCATE SPACE TO STORE DEVICE NAME
          50   82   9A  01B6   323          MOVZBL  (R2)+,R0                        ;GET LENGTH OF DEVICE NAME
               29   BB  01B9   324          PUSHR   #^M<R0,R3,R5>                   ;SAVE REGISTERS
```

```
                0C AE    62    50    28  01BB    325          MOVC    R0,(R2),12(SP)                  ;COPY DEVICE IDENTIFICATION
                         63    3A    90  01C0    326          MOVB    #^A/:/,(R3)                     ;APPEND THE :
                               29    BA  01C3    327          POPR    #^M<R0,R3,R5>                   ;RESTORE REGISTERS
                            7E    1B    9A  01C5    328          MOVZBL  #27,-(SP)                       ;INSERT ESCAPE AND TYPE ZERO
                          02 AE    53    B0  01C8    329          MOVW    R3,2(SP)                        ;INSERT INTERNAL FILE INDEX
                                               01CC    330
                                               01CC    331 ;
                                               01CC    332 ; BUILD ITEM LIST CONTAINING EQUIVALENCE NAME AND ATTRIBUTES.
                                               01CC    333 ;
                                  56    DD    01CC    334          PUSHL   R6                              ;SET THE ATTRIBUTES
                                  7E    7C    01CE    335          CLRQ    -(SP)                           ;TERMINATE THE ITEM LIST
                               0C AE    9F    01D0    336          PUSHAB  12(SP)                          ;SET ADDRESS OF EQUIV NAME
                               05 A0    9F    01D3    337          PUSHAB  5(R0)                           ;SET LENGTH OF EQUIV NAME
                          02 AE    02    B0  01D6    338          MOVW    #LNM$_STRING,2(SP)              ;SET ITEM CODE
                                  7E    D4  01DA    339          CLRL    -(SP)                           ;TERMINATE ATTRIBUTES ITEM
                             14 AE    9F    01DC    340          PUSHAB  20(SP)                          ;SET ATTRIBUTES
                         00030004 8F    DD    01DF    341          PUSHL   #LNM$_ATTRIBUTES@16+4           ;SET ITEM CODE AND LENGTH
                                               01E5    342
                                               01E5    343 ;
                                               01E5    344 ; PUSH NAME ATTRIBUTES AND ACCESS MODE, AND CREATE LOGICAL NAME AND TABLE
                                               01E5    345 ; NAME DESCRIPTORS.
                                               01E5    346 ;
                                  02    DD    01E5    347          PUSHL   #LNM$M_CONFINE                  ;PUSH NAME ATTRIBUTES
                                  02    DD    01E7    348          PUSHL   #PSL$C_SUPER                    ;PUSH ACCESS MODE
                               54    85    9A  01E9    349          MOVZBL  (R5)+,R4                        ;BUILD LOGICAL NAME DESCRIPTOR
                            7E    54    7D  01EC    350          MOVQ    R4,-(SP)                        ;
                          55    FE26  CF    9E  01EF    351          MOVAB   LNM$PROCESS,R5                  ;BUILD TABLE NAME DESCRIPTOR
                               54    85    9A  01F4    352          MOVZBL  (R5)+,R4                        ;
                            7E    54    7D  01F7    353          MOVQ    R4,-(SP)                        ;
                            51    5E    D0  01FA    354          MOVL    SP,R1                           ;SAVE ADDRESS OF DESCRIPTORS
                                               01FD    355
                                               01FD    356 ;
                                               01FD    357 ; PERFORM $CRELNM.
                                               01FD    358 ;
                                               01FD    359          $CRELNM_S  TABNAM=(R1),-                ;CREATE THE LOGICAL NAME
                                               01FD    360                     LOGNAM=8(R1),-               ;
                                               01FD    361                     ACMODE=16(R1),-              ;
                                               01FD    362                     ATTR=20(R1),-                ;
                                               01FD    363                     ITMLST=24(R1)                ;
                5E    0000004C 8F    C0    0212    364          ADDL    #<4*14>+20,SP                   ;CLEAN STACK
                                  05    0219    365          RSB                                     ;
                                               021A    366
                                               021A    367 ;
                                               021A    368 ; R5 = ADDRESS OF ASCIC LOGICAL NAME
                                               021A    369 ;
                                               021A    370 ; DELETE INPUT LOGICAL NAME.
                                               021A    371 ;
                                  02    DD    021A    372 DELETE: PUSHL   #PSL$C_SUPER                    ;PUSH ACCESS MODE
                               54    85    9A  021C    373          MOVZBL  (R5)+,R4                        ;BUILD LOGICAL NAME DESCRIPTOR
                            7E    54    7D  021F    374          MOVQ    R4,-(SP)                        ;
                          55    FDF3  CF    9E  0222    375          MOVAB   LNM$PROCESS,R5                  ;BUILD TABLE NAME DESCRIPTOR
                               54    85    9A  0227    376          MOVZBL  (R5)+,R4                        ;
                            7E    54    7D  022A    377          MOVQ    R4,-(SP)                        ;
                            51    5E    D0  022D    378          MOVL    SP,R1                           ;SAVE ADDRESS OF DESCRIPTORS
                                               0230    379          $DELLNM_S  TABNAM=(R1),-                ;DELETE THE LOGICAL NAME
                                               0230    380                     LOGNAM=8(R1),-               ;
                                               0230    381                     ACMODE=16(R1)                ;
```

```
                                                H 7
                     5E   14   C0  023F   382          ADDL    #4*5,SP                         ;CLEAN STACK
                          05   0242   383          RSB                                     ;
                               0243   384
                               0243   385          .END
```

```
$$.TMP1                         =  00000001
$$.TMP2                         =  00000069
CLI$_NORMAL                     =  00030001
CREATE                             000001B3 R      02
DCL$CREATE_IO                      00000167 RG     02
DCL$CREATE_OUTPUT                  00000193 RG     02
DCL$OPEN_CREATE                    ******** X      02
DCL$OPEN_OUTPUT                    00000025 RG     02
DCL$RESTORE_OUTPUT                 00000107 RG     02
DCL$VERIFY_IMAGE                   ******** X      02
DELETE                             0000021A R      02
FAB$B_FAC                       =  00000016
FAB$B_FNS                       =  00000034
FAB$B_ORG                       =  0000001D
FAB$B_RAT                       =  0000001E
FAB$B_RFM                       =  0000001F
FAB$B_SHR                       =  00000017
FAB$C_SEQ                       =  00000000
FAB$C_VFC                       =  00000003
FAB$L_DEV                       =  00000040
FAB$L_DNA                       =  00000030
FAB$L_FNA                       =  0000002C
FAB$L_FOP                       =  00000004
FAB$L_NAM                       =  00000028
FAB$M_CR                        =  00000002
FAB$M_PPF                       =  00040000
FAB$M_PPF_IND                   =  00004000
FAB$M_PRN                       =  00000004
FAB$M_PUT                       =  00000001
FAB$M_SQO                       =  00000040
FAB$W_IFI                       =  00000002
IDF_B_OUTFLAGS                     00000038
IDF_C_LENGTH                       00000074
IDF_K_LENGTH                       00000074
IDF_L_FILENAME                     00000068
IDF_L_INPRABCTX                    0000000C
IDF_L_LNK                          00000000
IDF_L_ONCTLY                       00000060
IDF_L_ONERROR                      00000008
IDF_L_OUTRABCTX                    00000024
IDF_L_SEARCHCTX                    00000064
IDF_Q_LABEL                        00000018
IDF_Q_LOCAL                        00000010
IDF_T_INPDVI                       0000003C
IDF_T_OUTDVI                       00000028
IDF_V_INPCCL                    =  00000001
IDF_V_OUTCCL                    =  00000000
IDF_W_FLAG                         0000005E
IDF_W_INPDID                       00000052
IDF_W_INPFID                       00000048
IDF_W_INPIFI                       00000004
IDF_W_INPRFA                       00000058
IDF_W_ONLEVEL                      00000006
IDF_W_OUTIFI                       00000020
IDF_W_OUTISI                       00000022
INPUTNAM                           00000000 R      02
LNM$M_CONFINE                   =  00000002
LNM$M_TERMINAL                  =  00000200
LNM$PROCESS                        00000019 R      02
LNM$V_CONCEALED                 =  00000008
LNM$_ATTRIBUTES                 =  00000003
LNM$_STRING                     =  00000002
NAM$L_FNB                       =  00000034
NAM$T_DVI                       =  00000014
NAM$V_CNCL_DEV                  =  0000000C
OUTFICE                            0000000A R      02
OUTPUTNAM                          0000000E R      02
OUT_B_OUTFLAGS                     00000018
OUT_C_LENGTH                       0000001C
OUT_K_LENGTH                       0000001C
OUT_L_OUTRABCTX                    00000004
OUT_T_OUTDVI                       00000008
OUT_V_OUTCCL                    =  00000000
OUT_W_OUTIFI                       00000000
OUT_W_OUTISI                       00000002
PRC_B_CONTINUE                     000000F3
PRC_B_DEFRADIX                     000000AE
PRC_B_EXMDEPMOD                    000000AD
PRC_B_EXMDEPWID                    000000AC
PRC_B_EXONLYL                      0000012D
PRC_B_FLAGS2                       000000AF
PRC_B_IMGFLAG                      00000078
PRC_B_OUTFLAGS                     0000012C
PRC_B_PROMPTLEN                    000000F0
PRC_C_LENGTH                       00000534
PRC_G_COMMANDS                     00000133
PRC_G_PROMPT                       000000F4
PRC_K_LENGTH                       00000534
PRC_L_CURRKEY                      00000048
PRC_L_EXMDEPADR                    000000A8
PRC_L_EXTARG                       00000094
PRC_L_EXTBLK                       0000008C
PRC_L_EXTCOD                       0000009C
PRC_L_EXTHND                       00000090
PRC_L_EXTPRM                       00000098
PRC_L_IDFLNK                       000000BC
PRC_L_IMGACTSTS                    00000080
PRC_L_INDCLOCK                     0000007C
PRC_L_INDEPTH                      0000005C
PRC_L_INDFAB                       0000001C
PRC_L_INDINPRAB                    00000014
PRC_L_INDOUTRAB                    00000018
PRC_L_INPRAB                       00000008
PRC_L_LASTKEY                      0000004C
PRC_L_LSTSTATUS                    000000B0
PRC_L_ONCTLY                       000000B8
PRC_L_ONERROR                      0000006C
PRC_L_OUTOFBAND                    000000B4
PRC_L_OUTRAB                       0000000C
PRC_L_OUTRABCTX                    00000118
PRC_L_PPFLIST                      00000070
PRC_L_RECALLPTR                    0000012F
PRC_L_RESTART                      00000058
PRC_L_SAVAP                        00000000
```

```
PRC_L_SAVFP                     00000004
PRC_L_SEVERITY                  00000050
PRC_L_SPWN                      000000C0
PRC_L_STACKLM                   000000A4
PRC_L_STACKPT                   000000A0
PRC_L_STATUS                    00000054
PRC_L_STS                       00000084
PRC_L_STV                       00000088
PRC_L_SYMBOL                    00000060
PRC_L_TMBX                      00000074
PRC_L_TRMLIST                   00000010
PRC_Q_ALLOCREG                  00000020
PRC_Q_COMMAND                   000000E0
PRC_Q_FLUSHTIME                 000000D0
PRC_Q_GLOBAL                    00000028
PRC_Q_IMAGENAME                 000000D8
PRC_Q_KEYPAD                    00000040
PRC_Q_LABEL                     00000030
PRC_Q_LOCAL                     00000038
PRC_Q_SAVEPRIV                  000000E8
PRC_T_OUTDVI                    0000011C
PRC_W_ASTIOSB                   000000C6
PRC_W_ASTRETN                   000000C8
PRC_W_ASTSTATUS                 000000C4
PRC_W_ATTMBX                    0000007A
PRC_W_FLAGS                     00000068
PRC_W_INPCHAN                   00000064
PRC_W_ONLEVEL                   0000006A
PRC_W_OUTIFI                    00000114
PRC_W_OUTISI                    00000116
PRC_W_OUTMBXCHN                 000000CA
PRC_W_OUTMBXREF                 000000CE
PRC_W_OUTMBXSIZ                 000000CC
PRC_W_PMPTCTRL                  000000F1
PRC_W_WAITIOSB                  00000066
PRD_C_LENGTH                    00000214
PRD_C_XLENGTH                   00000244
PRD_G_ALTINPRAB                 000000F4
PRD_G_ALTOUTRAB                 00000138
PRD_G_FAB                       00000000
PRD_G_INPRAB                    000000B0
PRD_G_NAM                       00000050
PRD_G_OUTRAB                    0000017C
PRD_G_TRMLIST                   000001E4
PRD_G_XABTRM                    000001C0
PRD_K_LENGTH                    00000214
PRD_K_XLENGTH                   00000244
PRD_T_OUTDVI                    00000214
PRD_T_OUTFNM                    00000230
PRD_W_OUTDID                    0000022A
PRD_W_OUTFID                    00000224
PSL$C_SUPER                 =   00000002
RAB$L_CTX                   =   00000018
RAB$L_FAB                   =   0000003C
RAB$S_PPF_RAT               =   00000008
RAB$V_PPF_RAT               =   00000006
RAB$W_ISI                   =   00000002
SYS$CLOSE                       ******** GX   02
SYS$CONNECT                     ******** GX   02
SYS$CRELNM                      ******** GX   02
SYS$DELLNM                      ******** GX   02
_$$_                        =   000000EF
```

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	00000534 (1332.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DCL$ZCODE	00000243 (579.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.07	00:00:00.50
Command processing	80	00:00:00.66	00:00:03.90
Pass 1	276	00:00:10.88	00:00:38.62
Symbol table sort	0	00:00:01.27	00:00:05.90
Pass 2	75	00:00:01.93	00:00:04.48
Symbol table output	20	00:00:00.17	00:00:00.62
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	462	00:00:15.00	00:00:54.04

The working set limit was 1350 pages.
53942 bytes (106 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 929 non-local and 11 local symbols.
385 source lines were read in Pass 1, producing 14 object records in Pass 2.
44 pages of virtual memory were used to define 31 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
_255DUA28:[DCL.OBJ]DCL.MLB;1	9
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	15
TOTALS (all libraries)	24

1158 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SYSOUTPUT/OBJ=OBJ$:SYSOUTPUT MSRC$:SYSOUTPUT/UPDATE=(ENH$:SYSOUTPUT)+EXECML$/LIB+LIB$:DCL/LIB+SYS$LIBRARY:SYSBLDMLB/L
```

PREFIX
REQ

STACLINT
LIS

STATUS
LIS

ANALYZE
LIS

DCXMSG
LIS

STASTUB
LIS

DCXDEF
MDL

EXPAND
LIS

SYSOUTPUT
LIS

DCXPRVDEF
MDL

STATEMENT
LIS

DCX

COMPRESS
LIS

TRANSFER
LIS

SYMBOL
LIS

DCXSHR
MAP

SUBS
LIS