```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

```
DDDDDDDD    HH      HH  EEEEEEEEEE    AAAAAA    DDDDDDDD      SSSSSSSS
DDDDDDDD    HH      HH  EEEEEEEEEE    AAAAAA    DDDDDDDD      SSSSSSSS
DD      DD  HH      HH  EE          AA      AA  DD      DD  SS
DD      DD  HH      HH  EE          AA      AA  DD      DD  SS
DD      DD  HH      HH  EE          AA      AA  DD      DD  SS
DD      DD  HH      HH  EE          AA      AA  DD      DD  SS
DD      DD  HHHHHHHHHH  EEEEEEEE    AA      AA  DD      DD    SSSSSS
DD      DD  HHHHHHHHHH  EEEEEEEE    AA      AA  DD      DD    SSSSSS
DD      DD  HH      HH  EE          AAAAAAAAAA  DD      DD          SS
DD      DD  HH      HH  EE          AAAAAAAAAA  DD      DD          SS
DD      DD  HH      HH  EE          AA      AA  DD      DD          SS      ....
DD      DD  HH      HH  EE          AA      AA  DD      DD          SS      ....
DDDDDDDD    HH      HH  EEEEEEEEEE  AA      AA  DDDDDDDD    SSSSSSSS        ....
DDDDDDDD    HH      HH  EEEEEEEEEE  AA      AA  DDDDDDDD    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
```

DQD

038
038
038
039
039
039
039
039
039
039
039
039
039
039
039
040
040
040
040
040
040
040
040
040
040
040
041
041
041
041
041
041
041
041
041
041
041
042
042
042
042
042
042
042
042
042
042
042
042
043
043
043
043
043
043
043
043
043
043
043
043
044
044
044
044

```
C
C Version:      'V04-000'
C
C*****************************************************************************
C*                                                                           *
C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
C*  ALL RIGHTS RESERVED.                                                     *
C*                                                                           *
C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
C*  TRANSFERRED.                                                             *
C*                                                                           *
C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
C*  CORPORATION.                                                             *
C*                                                                           *
C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
C*                                                                           *
C*                                                                           *
C*****************************************************************************
C


c       Author  Brian Porter            Creation Date   15-JUN-1981

c++
c       Functional description:
c
c       This routine is called to identify the device for 'DEVICE ERROR',
c       'DEVICE ATTENTION' and 'DEVICE TIMEOUT' entries.
c
c       Modified by:
c
C       V03-003 SAR0252         Sharon A. Reynolds      23-Apr-1984
C               Added a field length for the output of the device name.
C
C       V03-002 SAR0068         Sharon A. Reynolds,     20-Jun-1983
C               Changed the carriage control in the 'format' statements
C               for use with ERF.
C
c       v03-001 BP0003          Brian Porter,           03-MAR-1982
c               Added more arguments to dhead2 call.  Added dhead3
c               routine
c**
c--


        subroutine dhead1 (lun,device_type)
```

```
      include 'src$:msghdr.for /nolist'

      include 'src$:deverr.for /nolist'



      byte            lun

      character*(*)   device_type


      call header (lun)

      if (emb$w_hd_entry .eq. 1) then

      call logger (lun,'DEVICE ERROR')

      else if (emb$w_hd_entry .eq. 96) then

      call logger (lun,'DEVICE TIMEOUT')

      else if (emb$w_hd_entry .eq. 98) then

      call logger (lun,'DEVICE ATTENTION')
      endif

      call dhead2 (lun,device_type,emb$b_dv_namlng,emb$t_dv_name,
      1 emb$w_dv_unit)

      return

      end
```

```
PROGRAM SECTIONS

    Name                                  Bytes   Attributes

  0 $CODE                                   121   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                   45   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                  108   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 EMB                                     512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                   786

ENTRY POINTS

    Address  Type  Name

  0-00000000       DHEAD1

VARIABLES

    Address  Type  Name                                Address  Type  Name

 AP-00000008@ CHAR DEVICE_TYPE                       3-0000001C  L*1  EMB$B_DV_CLASS
  3-00000010  L*1  EMB$B_DV_ERTCNT                   3-00000011  L*1  EMB$B_DV_ERTMAX
  3-0000003E  L*1  EMB$B_DV_NAMLNG                   3-0000003A  L*1  EMB$B_DV_SLAVE
  3-0000001D  L*1  EMB$B_DV_TYPE                     3-00000036  I*4  EMB$L_DV_CHAR
  3-00000012  I*4  EMB$L_DV_IOSB1                    3-00000016  I*4  EMB$L_DV_IOSB2
  3-00000026  I*4  EMB$L_DV_MEDIA                    3-0000004E  I*4  EMB$L_DV_NUMREG
  3-0000002E  I*4  EMB$L_DV_OPCNT                    3-00000032  I*4  EMB$L_DV_OWNUIC
  3-0000001E  I*4  EMB$L_DV_RQPID                    3-00000000  I*4  EMB$L_HD_SID
  3-0000003F  CHAR EMB$T_DV_NAME                     3-00000024  I*2  EMB$W_DV_BCNT
  3-00000022  I*2  EMB$W_DV_BOFF                     3-0000002C  I*2  EMB$W_DV_ERRCNT
  3-0000003C  I*2  EMB$W_DV_FUNC                     3-0000001A  I*2  EMB$W_DV_STS
  3-0000002A  I*2  EMB$W_DV_UNIT                     3-00000004  I*2  EMB$W_HD_ENTRY
  3-0000000E  I*2  EMB$W_HD_ERRSEQ                 AP-00000004@ L*1  LUN

ARRAYS

    Address  Type  Name                                      Bytes  Dimensions

  3-00000000  L*1  EMB                                        512  (0:511)
  3-00000052  I*4  EMB$L_DV_REGSAV                            420  (0:104)
  3-00000006  I*4  EMB$Q_HD_TIME                                8  (2)

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name            Type  Name            Type  Name

        DHEAD2                HEADER                LOGGER
```

```
0001
0002
0003
0004
0005            subroutine dhead2 (lun,device_type,emb$b_zz_namlng,emb$t_zz_name,
0006            1 emb$w_zz_unit)
0007
0008
0009
0010
0011            include 'src$:msghdr.for /nolist'
0070
0071
0072
0073
0074            byte            lun
0075
0076            character*(*)   device_type
0077
0078            byte            emb$b_zz_namlng
0079
0080            character*(*)   emb$t_zz_name
0081
0082            integer*2       emb$w_zz_unit
0083
0084            integer*4       caller_mount_flag_and_label
0085
0086            character*12    last_emb$t_vm_label
0087
0088            integer*4       last_emb$t_vm_label_length
0089
0090            logical*1       last_emb$t_vm_label_valid
0091
0092            logical*1       str$trim
0093
0094            integer*4       compress4
0095
0096
0097
0098
0099            last_emb$t_vm_label_valid = .false.
0100
0101            call get_current_label (3,emb$l_hd_sid,emb$b_zz_namlng,emb$t_zz_name,
0102            1 emb$w_zz_unit,%ref(last_emb$t_vm_label),*5)
0103
0104            last_emb$t_vm_label_valid = .true.
0105
0106    5       continue
0107
0108            if (.not. str$trim (last_emb$t_vm_label,last_emb$t_vm_label,
0109            1 last_emb$t_vm_label_length)) then
0110
0111            last_emb$t_vm_label_length = 12
0112            endif
0113
0114            if (
0115            1 last_emb$t_vm_label_valid
```

```
     1 .and.
     1 last_emb$t_vm_label_length .gt. 1
     1 ) then

10   continue

     call linchk (lun,2)

     write(lun,15) device_type,emb$t_zz_name(1:emb$b_zz_namlng),
     1 emb$w_zz_unit,', CURRENT LABEL ''',
     1 last_emb$t_vm_label(1:last_emb$t_vm_label_length),''''
15   format(/' ',a,' SUB-SYSTEM, UNIT ',a<emb$b_zz_namlng>,
     1 i<compress4 (lib$extzv(0,16,emb$w_zz_unit))>,':',3(:a))

     return
     endif

20   continue

     call linchk (lun,2)

     write(lun,15) device_type,emb$t_zz_name(1:emb$b_zz_namlng),
     1 emb$w_zz_unit

     return


     entry dhead3 (lun,device_type,emb$b_zz_namlng,emb$t_zz_name,
     1 emb$w_zz_unit,caller_mount_flag_and_label)


     last_emb$t_vm_label_valid = .false.

     if (caller_mount_flag_and_label .eq. -1) goto 20

     call movc3 (%val(12),%val(caller_mount_flag_and_label),
     1 %ref(last_emb$t_vm_label))

     if (.not. str$trim (last_emb$t_vm_label,last_emb$t_vm_label,
     1 last_emb$t_vm_label_length)) then

     last_emb$t_vm_label_length = 12
     endif

     last_emb$t_vm_label_valid = .true.

     goto 10

     end
```

```
PROGRAM SECTIONS

    Name                                  Bytes   Attributes

  0 $CODE                                   449   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                   83   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                  180   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 EMB                                     512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                  1224

ENTRY POINTS

    Address  Type  Name              Address  Type  Name

  0-00000000       DHEAD2          0-0000012B       DHEAD3

VARIABLES

    Address  Type  Name                                Address  Type  Name

 AP-00000018@ I*4  CALLER_MOUNT_FLAG_AND_LABEL       AP-00000008@ CHAR DEVICE_TYPE
  2-00000014@ L*1  EMB$B_ZZ_NAMLNG                    3-00000000  I*4  EMB$L_RD_SID
 AP-00000010@ CHAR EMB$T_ZZ_NAME                      3-00000004  I*2  EMB$W_HD_ENTRY
  3-0000000E  I*2  EMB$W_HD_ERRSEQ                    2-00000018@ I*2  EMB$W_ZZ_UNIT
  2-00000000  CHAR LAST_EMB$T_VM_LABEL                2-00000010  I*4  LAST_EMB$T_VM_LABEL_LENGTH
  2-0000000C  L*1  LAST_EMB$T_VM_LABEL_VALID         AP-00000004@ L*1  LUN

ARRAYS

    Address  Type  Name                  Bytes  Dimensions

  3-00000000  L*1  EMB                     512  (0:511)
  3-00000006  I*4  EMB$Q_HD_TIME             8  (2)

LABELS

    Address   Label       Address   Label       Address   Label       Address   Label

  0-0000004A  5         0-00000062  10        1-00000022  15'       0-000000DA  20

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                              Type  Name                              Type  Name

   I*4  COMPRESS4                                GET_CURRENT_LABEL                  I*4  LIB$EXTZV
        LINCHK                                   MOVC3                              L*1  STR$TRIM
```

```
COMMAND QUALIFIERS

  FORTRAN /LIS=LIS$:DHEADS/OBJ=OBJ$:DHEADS MSRC$:DHEADS

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
  /F77  /NOG_FLOATING  /I4  /OPTIMIZE  /WARNINGS  /NOD_LINES  /NOCROSS_REFERENCE  /NOMACHINE_CODE  /CONTINUATIONS=19


COMPILATION STATISTICS

  Run Time:           2.96 seconds
  Elapsed Time:       9.20 seconds
  Page Faults:        132
  Dynamic Memory:     175 pages
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CLASSIFY
LIS

DR750
LIS

CSTRING
LIS

DHEADS
LIS

DR780
LIS

DR11W
LIS

COMPRESS
LIS

DTAILS
LIS

DECODECC
LIS

DUMPREG
LIS

DUTLDRIVR
LIS

CALCMAP
LIS

DUP3271
LIS

CRYPTK
LIS

DQDISKS
LIS

DUP11
LIS