```
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
```

FILEID**MASSTAPE

```
MM      MM    AAAAAA      SSSSSSSS    SSSSSSSS  TTTTTTTTTT    AAAAAA    PPPPPPPP    EEEEEEEEEE
MM      MM    AAAAAA      SSSSSSSS    SSSSSSSS  TTTTTTTTTT    AAAAAA    PPPPPPPP    EEEEEEEEEE
MMMM  MMMM  AA      AA  SS          SS            TT      AA      AA  PP      PP  EE
MMMM  MMMM  AA      AA  SS          SS            TT      AA      AA  PP      PP  EE
MM  MM  MM  AA      AA  SS          SS            TT      AA      AA  PP      PP  EE
MM  MM  MM  AA      AA  SS          SS            TT      AA      AA  PP      PP  EE
MM      MM  AA      AA    SSSSSS      SSSSSS      TT      AA      AA  PPPPPPPP    EEEEEEEE
MM      MM  AA      AA    SSSSSS      SSSSSS      TT      AA      AA  PPPPPPPP    EEEEEEEE
MM      MM  AAAAAAAAAA          SS          SS    TT      AAAAAAAAAA  PP          EE
MM      MM  AAAAAAAAAA          SS          SS    TT      AAAAAAAAAA  PP          EE
MM      MM  AA      AA          SS          SS    TT      AA      AA  PP          EE              ....
MM      MM  AA      AA          SS          SS    TT      AA      AA  PP          EE              ....
MM      MM  AA      AA  SSSSSSSS    SSSSSSSS      TT      AA      AA  PP          EEEEEEEEEE      ....
MM      MM  AA      AA  SSSSSSSS    SSSSSSSS      TT      AA      AA  PP          EEEEEEEEEE      ....

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
```

```
0001            SUBROUTINE MASSTAPE (LUN)
0002    C
0003    C Version:      'V04-000'
0004    C
0005    C*****************************************************************************
0006    C*                                                                           *
0007    C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0008    C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0009    C*  ALL RIGHTS RESERVED.                                                     *
0010    C*                                                                           *
0011    C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0012    C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0013    C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0014    C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0015    C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0016    C*  TRANSFERRED.                                                             *
0017    C*                                                                           *
0018    C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0019    C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0020    C*  CORPORATION.                                                             *
0021    C*                                                                           *
0022    C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0023    C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0024    C*                                                                           *
0025    C*                                                                           *
0026    C*****************************************************************************
0027    C
0028    C
0029    C
0030    C       AUTHOR  BRIAN PORTER            CREATION DATE   26-MAR-1979
0031    C
0032    C       Functional description:
0033    C
0034    C       This module displays error log entries to all drive types connected
0035    C       to the TM03 formatter.
0036    C
0037    C       Modified by:
0038    C
0039    C       V03-004 SAR0225         Sharon A. Reynolds,             28-Mar-1984
0040    C               Changed the call to UCB$L_OWNUIC to ORB$L_OWNER.
0041    C
0042    C       V03-003 SAR0079         Sharon A. Reynolds,             20-Jun-1983
0043    C               Changed the carriage control in the 'format' statements
0044    C               for use with ERF.
0045    C
0046    C       V03-002 SAR0043         Sharon A. Reynolds,             9-Jun-1983
0047    C               Removed brief/cryptic support.
0048    C
0049    C       v03-001 BP0001          Brian Porter,                   20-AUG-1982
0050    C               Minor edit.
0051    C**
0052
0053            INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
0112            INCLUDE 'SRC$:DEVERR.FOR /NOLIST'
0213
0214
0215            BYTE            LUN
```

```
0216
0217            INTEGER*4       FIELD
0218            INTEGER*4       COMPRESSC
0219            integer*4       selected_mapping_register
0220            integer*4       adapter_registers(7)
0221            INTEGER*4       DRIVE_CS1
0222            INTEGER*4       DRIVE_DS
0223            INTEGER*4       DRIVE_ER
0224            INTEGER*4       DRIVE_MR
0225            INTEGER*4       DRIVE_AS
0226            INTEGER*4       DRIVE_FC
0227            INTEGER*4       DRIVE_DT
0228            INTEGER*4       DRIVE_CK
0229            INTEGER*4       DRIVE_SN
0230            INTEGER*4       DRIVE_TC
0231            INTEGER*4       DRIVE_FUNC
0232
0233            equivalence     (adapter_registers,emb$l_dv_regsav(0))
0234            EQUIVALENCE     (DRIVE_CS1,EMB$L_DV_REGSAV(7))
0235            EQUIVALENCE     (DRIVE_DS,EMB$L_DV_REGSAV(8))
0236            EQUIVALENCE     (DRIVE_ER,EMB$L_DV_REGSAV(9))
0237            EQUIVALENCE     (DRIVE_MR,EMB$L_DV_REGSAV(10))
0238            EQUIVALENCE     (DRIVE_AS,EMB$L_DV_REGSAV(11))
0239            EQUIVALENCE     (DRIVE_FC,EMB$L_DV_REGSAV(12))
0240            EQUIVALENCE     (DRIVE_DT,EMB$L_DV_REGSAV(13))
0241            EQUIVALENCE     (DRIVE_CK,EMB$L_DV_REGSAV(14))
0242            EQUIVALENCE     (DRIVE_SN,EMB$L_DV_REGSAV(15))
0243            EQUIVALENCE     (DRIVE_TC,EMB$L_DV_REGSAV(16))
0244
0245            logical*1       diagnostic_mode
0246
0247            PARAMETER       TIMEOUT  = 96
0248            PARAMETER       XFER_CMD = 20
0249            PARAMETER       1600_BPI = 1
0250
0251            CHARACTER*26    MASS_ER7(0:1)
0252            CHARACTER*18    MASS_ER10(0:1)
0253            CHARACTER*23    MASS_ER15(0:1)
0254
0255
0256            CHARACTER*7     MASS1_CS1(0:0)
0257            DATA    MASS1_CS1(0)    /'GO BIT*'/
0258
0259            CHARACTER*16    MASS2_CS1(11:11)
0260            DATA    MASS2_CS1(11)   /'DRIVE AVAILABLE*'/
0261
0262            CHARACTER*24    MASS1_DS(0:8)
0263            DATA    MASS1_DS(0)     /'TRANSPORT ATTENTION*'/
0264            DATA    MASS1_DS(1)     /'BEGINNING OF TAPE*'/
0265            DATA    MASS1_DS(2)     /'TAPE MARK*'/
0266            DATA    MASS1_DS(3)     /'IDENTIFICATION BURST*'/
0267            DATA    MASS1_DS(4)     /'SETTLE DOWN*'/
0268            DATA    MASS1_DS(5)     /'PHASE ENCODED STATUS*'/
0269            DATA    MASS1_DS(6)     /'TRANSPORT STATUS CHANGE*'/
0270            DATA    MASS1_DS(7)     /'DRIVE READY*'/
0271            DATA    MASS1_DS(8)     /'DRIVE PRESENT*'/
0272
```

```
0273            CHARACTER*24    MASS2_DS(10:15)
0274            DATA    MASS2_DS(10)    /'END OF TAPE*'/
0275            DATA    MASS2_DS(11)    /'WRITE PROTECTED*'/
0276            DATA    MASS2_DS(12)    /'MEDIUM ON-LINE*'/
0277            DATA    MASS2_DS(13)    /'POSITIONING IN PROGRESS*'/
0278            DATA    MASS2_DS(14)    /'COMPOSITE ERROR*'/
0279            DATA    MASS2_DS(15)    /'ATTENTION ACTIVE*'/
0280
0281            CHARACTER*30    MASS1_ER(0:5)
0282            DATA    MASS1_ER(0)     /'ILLEGAL FUNCTION*'/
0283            DATA    MASS1_ER(1)     /'ILLEGAL REGISTER*'/
0284            DATA    MASS1_ER(2)     /'REGISTER MODIFICATION REFUSED*'/
0285            DATA    MASS1_ER(3)     /'MASSBUS CONTROL PARITY ERROR*'/
0286            DATA    MASS1_ER(4)     /'FORMAT CODE ERROR*'/
0287            DATA    MASS1_ER(5)     /'MASSBUS DATA PARITY ERROR*'/
0288
0289            CHARACTER*25    PE1_ER(6:7)
0290            DATA    PE1_ER(6)       /'UNCORRECTABLE DATA ERROR*'/
0291            DATA    PE1_ER(7)       /'FORMAT ERROR*'/
0292
0293            CHARACTER*26    NRZI1_ER(6:7)
0294            DATA    NRZI1_ER(6)     /'VERTICAL PARITY ERROR*'/
0295            DATA    NRZI1_ER(7)     /'LONGITUDINAL PARITY ERROR*'/
0296
0297            CHARACTER*18    MASS2_ER(8:9)
0298            DATA    MASS2_ER(8)     /'NON-STANDARD GAP*'/
0299            DATA    MASS2_ER(9)     /'FRAME COUNT ERROR*'/
0300
0301            CHARACTER*17    PE2_ER(10:10)
0302            DATA    PE2_ER(10)      /'CORRECTABLE SKEW*'/
0303
0304            CHARACTER*18    NRZI2_ER(10:10)
0305            DATA    NRZI2_ER(10)    /'ILLEGAL TAPE MARK*'/
0306
0307            CHARACTER*24    MASS3_ER(11:14)
0308            DATA    MASS3_ER(11)    /'NON-EXECUTABLE FUNCTION*'/
0309            DATA    MASS3_ER(12)    /'DRIVE TIMING ERROR*'/
0310            DATA    MASS3_ER(13)    /'OPERATION INCOMPLETE*'/
0311            DATA    MASS3_ER(14)    /'TRANSPORT UNSAFE*'/
0312
0313            CHARACTER*23    PE3_ER(15:15)
0314            DATA    PE3_ER(15)      /'CORRECTABLE DATA ERROR*'/
0315
0316            CHARACTER*10    NRZI3_ER(15:15)
0317            DATA    NRZI3_ER(15)    /'CRC ERROR*'/
0318
0319            CHARACTER*27    MASS1_DT(0:3)
0320            DATA    MASS1_DT(0)     /'45 IPS TRANSPORT SELECTED*'/
0321            DATA    MASS1_DT(1)     /'75 IPS TRANSPORT SELECTED*'/
0322            DATA    MASS1_DT(2)     /'125 IPS TRANSPORT SELECTED*'/
0323            DATA    MASS1_DT(3)     /'BIT 3 ALWAYS SET*'/
0324
0325            CHARACTER*15    MASS_DT5(0:1)
0326            DATA    MASS_DT5(0)     /'TM02 FORMATTER*'/
0327            DATA    MASS_DT5(1)     /'TM03 FORMATTER*'/
0328
0329            CHARACTER*18    MASS2_DT(10:10)
```

```
0330            DATA    MASS2_DT(10)    /'TRANSPORT PRESENT*'/
0331
0332            CHARACTER*10    MASS3_DT(12:12)
0333            DATA    MASS3_DT(12)    /'7-CHANNEL*'/
0334
0335            CHARACTER*23    MASS4_DT(14:15)
0336            DATA    MASS4_DT(14)    /'TAPE DRIVE*'/
0337            DATA    MASS4_DT(15)    /'NOT SECTOR ADDRESSABLE*'/
0338
0339            CHARACTER*12    MASS1_TC(3:3)
0340            DATA    MASS1_TC(3)     /'EVEN PARITY*'/
0341
0342            CHARACTER*15    MASS1_FORMAT(0:2)
0343            DATA    MASS1_FORMAT(0) /'10-CORE DUMP*'/
0344            DATA    MASS1_FORMAT(1) /'15-CORE DUMP*'/
0345            DATA    MASS1_FORMAT(2) /'10-COMPATIBLE*'/
0346
0347            CHARACTER*13    MASS2_FORMAT(12:14)
0348            DATA    MASS2_FORMAT(12)/'11-NORMAL*'/
0349            DATA    MASS2_FORMAT(13)/'11-CORE DUMP*'/
0350            DATA    MASS2_FORMAT(14)/'15-NORMAL*'/
0351
0352            CHARACTER*9     MASS_DENSITY(3:4)
0353            DATA    MASS_DENSITY(3)/'800 NRZI*'/
0354            DATA    MASS_DENSITY(4)/'1600 PE*'/
0355
0356            CHARACTER*32    MASS2_TC(12:15)
0357            DATA    MASS2_TC(12)    /'ENABLE ABORT ON DATA XFER ERROR*'/
0358            DATA    MASS2_TC(13)    /'TRANSPORT ADDRESS CHANGE*'/
0359            DATA    MASS2_TC(14)    /'FRAME COUNT STATUS*'/
0360            DATA    MASS2_TC(15)    /'ACCELERATION*'/
0361
0362            CHARACTER*24    MASS_FUNC(0:31)
0363            DATA    MASS_FUNC(0)    /'NO-OPERATION*'/
0364            DATA    MASS_FUNC(1)    /'REWIND AND SET OFF-LINE*'/
0365            DATA    MASS_FUNC(2)    /'ILLEGAL FUNCTION*'/
0366            DATA    MASS_FUNC(3)    /'REWIND*'/
0367            DATA    MASS_FUNC(4)    /'DRIVE CLEAR*'/
0368            DATA    MASS_FUNC(5)    /'ILLEGAL FUNCTION*'/
0369            DATA    MASS_FUNC(6)    /'ILLEGAL FUNCTION*'/
0370            DATA    MASS_FUNC(7)    /'ILLEGAL FUNCTION*'/
0371            DATA    MASS_FUNC(8)    /'READ-IN PRESET*'/
0372            DATA    MASS_FUNC(9)    /'ILLEGAL FUNCTION*'/
0373            DATA    MASS_FUNC(10)   /'ERASE*'/
0374            DATA    MASS_FUNC(11)   /'WRITE TAPE MARK*'/
0375            DATA    MASS_FUNC(12)   /'SPACE FORWARD*'/
0376            DATA    MASS_FUNC(13)   /'SPACE REVERSE*'/
0377            DATA    MASS_FUNC(14)   /'ILLEGAL FUNCTION*'/
0378            DATA    MASS_FUNC(15)   /'ILLEGAL FUNCTION*'/
0379            DATA    MASS_FUNC(16)   /'ILLEGAL FUNCTION*'/
0380            DATA    MASS_FUNC(17)   /'ILLEGAL FUNCTION*'/
0381            DATA    MASS_FUNC(18)   /'ILLEGAL FUNCTION*'/
0382            DATA    MASS_FUNC(19)   /'ILLEGAL FUNCTION*'/
0383            DATA    MASS_FUNC(20)   /'WRITE CHECK FORWARD*'/
0384            DATA    MASS_FUNC(21)   /'ILLEGAL FUNCTION*'/
0385            DATA    MASS_FUNC(22)   /'ILLEGAL FUNCTION*'/
0386            DATA    MASS_FUNC(23)   /'WRITE CHECK REVERSE*'/
```

```
0387            DATA    MASS_FUNC(24)   /'WRITE FORWARD*'/
0388            DATA    MASS_FUNC(25)   /'ILLEGAL FUNCTION*'/
0389            DATA    MASS_FUNC(26)   /'ILLEGAL FUNCTION*'/
0390            DATA    MASS_FUNC(27)   /'ILLEGAL FUNCTION*'/
0391            DATA    MASS_FUNC(28)   /'READ FORWARD*'/
0392            DATA    MASS_FUNC(29)   /'ILLEGAL FUNCTION*'/
0393            DATA    MASS_FUNC(30)   /'ILLEGAL FUNCTION*'/
0394            DATA    MASS_FUNC(31)   /'READ REVERSE*'/
0395
0396
0397            CALL FRCTOF (LUN)
0398
0399            call dhead1 (lun,'MASSBUS')
0400
0401            DRIVE_FUNC=LIB$EXTZV(1,5,DRIVE_CS1)
0402
0403            if (
0404            1 emb$w_dv_bcnt .ne. 0
0405            1 .and.
0406            1 drive_func .ge. xfer_cmd
0407            1 .and.
0408            1 emb$w_hd_entry .ne. 98
0409            1 ) then
0410
0411            call mba_control_registers (lun,5,adapter_registers,
0412            1 selected_mapping_register)
0413
0414            call mba_mapping_register (lun,selected_mapping_register,
0415            1 adapter_registers(6))
0416
0417            if (selected_mapping_register .gt. 0) then
0418
0419            call mba_mapping_register (lun,(selected_mapping_register - 1),
0420            1 adapter_registers(7))
0421            endif
0422            endif
0423
0424            diagnostic_mode = .false.
0425
0426            if (lib$extzv(0,1,drive_mr) .eq. 1) diagnostic_mode = .true.
0427
0428    C
0429    C       'CS1' REGISTER
0430    C
0431
0432            CALL LINCHK (LUN,2)
0433
0434            WRITE(LUN,75) DRIVE_CS1
0435    75      FORMAT(/' ',T8,'TMCS1',T24,Z8.8)
0436
0437            if (.not. diagnostic_mode) then
0438
0439            call mba_status_register16_31 (lun,drive_cs1,drive_cs1,0)
0440
0441            CALL OUTPUT (LUN,DRIVE_CS1,MASS1_CS1,0,0,0,'0')
0442
0443            CALL LINCHK (LUN,1)
```

```
0444
0445            FIELD=LIB$EXTZV(1,5,DRIVE_CS1)
0446
0447            WRITE(LUN,100) MASS_FUNC(FIELD)
0448    100     FORMAT(' ',T40,A<COMPRESSC (MASS_FUNC(FIELD))>)
0449
0450            CALL OUTPUT (LUN,DRIVE_CS1,MASS2_CS1,11,11,11,'0')
0451            endif
0452
0453    C
0454    C       'DS' REGISTER
0455    C
0456
0457            CALL LINCHK (LUN,1)
0458
0459            WRITE(LUN,125) DRIVE_DS
0460    125     FORMAT(' ',T8,'TMDS',T24,Z8.8)
0461
0462            if (.not. diagnostic_mode) then
0463
0464            call mba_status_register16_31 (lun,drive_cs1,drive_ds,1)
0465
0466            CALL OUTPUT (LUN,DRIVE_DS,MASS1_DS,0,0,8,'0')
0467
0468            CALL OUTPUT (LUN,DRIVE_DS,MASS2_DS,10,10,15,'0')
0469            endif
0470
0471    C
0472    C       'ER' REGISTER
0473    C
0474
0475            CALL LINCHK (LUN,1)
0476
0477            WRITE(LUN,140) DRIVE_ER
0478    140     FORMAT(' ',T8,'TMER',T24,Z8.8)
0479
0480            if (.not. diagnostic_mode) then
0481
0482            call mba_status_register16_31 (lun,drive_ds,drive_er,1)
0483
0484            CALL OUTPUT (LUN,DRIVE_ER,MASS1_ER,0,0,5,'0')
0485
0486            FIELD=LIB$EXTZV(10,1,DRIVE_TC)
0487
0488            IF (FIELD .EQ. 1600_BPI) THEN
0489
0490            CALL OUTPUT (LUN,DRIVE_ER,PE1_ER,6,6,7,'0')
0491            ELSE
0492
0493            CALL OUTPUT (LUN,DRIVE_ER,NRZI1_ER,6,6,7,'0')
0494            ENDIF
0495
0496            CALL OUTPUT (LUN,DRIVE_ER,MASS2_ER,8,8,9,'0')
0497
0498            IF (FIELD .EQ. 1600_BPI) THEN
0499
0500            CALL OUTPUT (LUN,DRIVE_ER,PE2_ER,10,10,10,'0')
```

```
0501            ELSE
0502
0503            CALL OUTPUT (LUN,DRIVE_ER,NRZI2_ER,10,10,10,'0')
0504            ENDIF
0505
0506            CALL OUTPUT (LUN,DRIVE_ER,MASS3_ER,11,11,14,'0')
0507
0508            IF (FIELD .EQ. 1600_BPI) THEN
0509
0510            CALL OUTPUT (LUN,DRIVE_ER,PE3_ER,15,15,15,'0')
0511            ELSE
0512
0513            CALL OUTPUT (LUN,DRIVE_ER,NRZI3_ER,15,15,15,'0')
0514            ENDIF
0515            endif
0516
0517    C
0518    C       'MR' REGISTER
0519    C
0520
0521            CALL LINCHK (LUN,1)
0522
0523            WRITE(LUN,150) DRIVE_MR
0524    150     FORMAT(' ',T8,'TMMR',T24,Z8.8)
0525
0526            if (.not. diagnostic_mode) then
0527
0528            call mba_status_register16_31 (lun,drive_er,drive_mr,1)
0529            else
0530
0531            CALL LINCHK (LUN,1)
0532
0533            WRITE(LUN,160)
0534    160     FORMAT(' ',T40,'DIAGNOSTIC MODE')
0535            endif
0536
0537    C
0538    C       'AS' REGISTER
0539    C
0540
0541            CALL LINCHK (LUN,1)
0542
0543            WRITE(LUN,175) DRIVE_AS
0544    175     FORMAT(' ',T8,'TMAS',T24,Z8.8)
0545
0546            if (.not. diagnostic_mode) then
0547
0548            call mba_status_register16_31 (lun,drive_mr,drive_as,1)
0549
0550            DO 184 I=0,7
0551
0552            if (lib$extzv(i,1,drive_as) .eq. 1) then
0553
0554            CALL LINCHK (LUN,1)
0555
0556            WRITE(LUN,183) I
0557    183     FORMAT(' ',T40,'ATTENTION UNIT ',I1,'.')
```

```
0558                endif
0559
0560    184         CONTINUE
0561                endif
0562
0563        C
0564        C       'FC' REGISTER
0565        C
0566
0567                CALL LINCHK (LUN,1)
0568
0569                WRITE(LUN,185) DRIVE_FC
0570    185         FORMAT(' ',T8,'TMFC',T24,Z8.8)
0571
0572                if (.not. diagnostic_mode) then
0573
0574                call mba_status_register16_31 (lun,drive_as,drive_fc,1)
0575                endif
0576
0577        C
0578        C       'DT' REGISTER
0579        C
0580
0581                CALL LINCHK (LUN,1)
0582
0583                WRITE(LUN,200) DRIVE_DT
0584    200         FORMAT(' ',T8,'TMDT',T24,Z8.8)
0585
0586                if (.not. diagnostic_mode) then
0587
0588                call mba_status_register16_31 (lun,drive_fc,drive_dt,1)
0589
0590                FIELD=LIB$EXTZV(0,3,DRIVE_DT)
0591
0592                IF (FIELD .EQ. 0) THEN
0593
0594                CALL LINCHK (LUN,1)
0595
0596                WRITE(LUN,215)
0597    215         FORMAT(' ',T40,'TRANSPORT NOT SELECTED')
0598                ELSE
0599
0600                CALL OUTPUT (LUN,DRIVE_DT,MASS1_DT,0,0,3,'0')
0601                ENDIF
0602
0603                CALL LINCHK (LUN,1)
0604
0605                FIELD=LIB$EXTZV(5,1,DRIVE_DT)
0606
0607                WRITE(LUN,225) MASS_DT5(FIELD)
0608    225         FORMAT(' ',T40,A<COMPRESSC (MASS_DT5(FIELD))>)
0609
0610                CALL OUTPUT (LUN,DRIVE_DT,MASS2_DT,10,10,10,'0')
0611
0612                CALL OUTPUT (LUN,DRIVE_DT,MASS3_DT,12,12,12,'0')
0613
0614                CALL OUTPUT (LUN,DRIVE_DT,MASS4_DT,14,14,15,'0')
```

```
0615            endif
0616
0617    C
0618    C       'CK' REGISTER
0619    C
0620
0621            CALL LINCHK (LUN,1)
0622
0623            WRITE(LUN,325) DRIVE_CK
0624    325     FORMAT(' ',T8,'TMCK',T24,Z8.8)
0625
0626            if (.not. diagnostic_mode) then
0627
0628            call mba_status_register16_31 (lun,drive_dt,drive_ck,1)
0629
0630            FIELD=LIB$EXTZV(10,1,DRIVE_TC)
0631
0632            IF (FIELD .EQ. I600_BPI) THEN
0633
0634            DO 340 I=0,8
0635
0636            if (lib$extzv(i,1,drive_ck) .eq. 1) then
0637
0638            CALL LINCHK (LUN,1)
0639
0640            IF (I .LE. 7) THEN
0641
0642            WRITE(LUN,330) I
0643    330     FORMAT(' ',T40,'DEAD TRACK CHANNEL ',I1,'.')
0644            ELSE
0645
0646            WRITE(LUN,335)
0647    335     FORMAT(' ',T40,'DEAD TRACK PARITY CHANNEL')
0648            ENDIF
0649            endif
0650
0651    340     CONTINUE
0652            ENDIF
0653            endif
0654
0655    C
0656    C       'SN' REGISTER
0657    C
0658
0659            CALL LINCHK (LUN,1)
0660
0661            WRITE(LUN,350) DRIVE_SN
0662    350     FORMAT(' ',T8,'TMSN',T24,Z8.8)
0663
0664            if (.not. diagnostic_mode) then
0665
0666            call mba_status_register16_31 (lun,drive_ck,drive_sn,1)
0667            endif
0668
0669    C
0670    C       'TC' REGISTER
0671    C
```

```
0672
0673            CALL LINCHK (LUN,1)
0674
0675            WRITE(LUN,375) DRIVE_TC
0676    375     FORMAT(' ',T8,'TMTC',T24,Z8.8)
0677
0678            if (.not. diagnostic_mode) then
0679
0680            call mba_status_register16_31 (lun,drive_sn,drive_tc,1)
0681
0682            CALL LINCHK (LUN,1)
0683
0684            FIELD=LIB$EXTZV(0,3,DRIVE_TC)
0685
0686            WRITE(LUN,400) FIELD
0687    400     FORMAT(' ',T40,'TRANSPORT SELECT = ',I1,'.')
0688
0689            CALL LINCHK (LUN,1)
0690
0691            FIELD=LIB$EXTZV(4,4,DRIVE_TC)
0692
0693            IF (FIELD .LE. 2) THEN
0694
0695            WRITE(LUN,425) MASS1_FORMAT(FIELD)
0696    425     FORMAT(' ',T40,A<COMPRESSC (MASS1_FORMAT(FIELD))>)
0697
0698            ELSE IF (FIELD .GE. 12 .AND. FIELD .LE. 14) THEN
0699
0700            WRITE(LUN,450) MASS2_FORMAT(FIELD)
0701    450     FORMAT(' ',T40,'FORMAT  = ',
0702            1 A<COMPRESSC (MASS2_FORMAT(FIELD))>)
0703            ENDIF
0704
0705            CALL LINCHK (LUN,1)
0706
0707            FIELD=LIB$EXTZV(8,3,DRIVE_TC)
0708
0709            IF (FIELD .EQ. 3 .OR. FIELD .EQ. 4) THEN
0710
0711            WRITE(LUN,500) MASS_DENSITY(FIELD)
0712    500     FORMAT(' ',T40,'DENSITY = ',
0713            1 A<COMPRESSC (MASS_DENSITY(FIELD))>)
0714            ENDIF
0715
0716            CALL OUTPUT (LUN,DRIVE_TC,MASS2_TC,12,12,15,'0')
0717            endif
0718
0719            call linchk (lun,1)
0720
0721            write(lun,530)
0722    530     format(' ',:)
0723
0724            if (emb$w_hd_entry .ne. 98) then
0725
0726            call ucb$b_ertcnt (lun,emb$b_dv_ertcnt)
0727
0728            call ucb$b_ertmax (lun,emb$b_dv_ertmax)
```

```
0729            endif
0730
0731            call orb$l_owner (lun,emb$l_dv_ownuic)
0732
0733            call ucb$l_char (lun,emb$l_dv_char)
0734
0735            call ucb$w_sts (lun,emb$w_dv_sts)
0736
0737            call ucb$l_opcnt (lun,emb$l_dv_opcnt)
0738
0739            call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0740
0741            if (emb$w_hd_entry .ne. 98) then
0742
0743            call linchk (lun,1)
0744
0745            write(lun,530)
0746
0747            call masstape_qio (lun,emb$w_dv_func)
0748
0749            call irp$w_bcnt (lun,emb$w_dv_bcnt)
0750
0751            call irp$w_boff (lun,emb$w_dv_boff)
0752
0753            call irp$l_pid (lun,emb$l_dv_rqpid)
0754
0755            call irp$q_iosb (lun,emb$l_dv_iosb1)
0756            endif
0757
0758            RETURN
0759            END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	2347	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	509	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	4072	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	EMB	512	PIC OVR REL CBL SHR NOEXE RD WRT LONG
	Total Space Allocated	7440	

ENTRY POINTS

Address	Type	Name
0-00000000		MASSTAPE

VARIABLES

Address	Type	Name	Address	Type	Name
2-000008A7	L*1	DIAGNOSTIC_MODE	3-0000007E	I*4	DRIVE_AS
3-0000008A	I*4	DRIVE_CK	3-0000006E	I*4	DRIVE_CS1
3-00000072	I*4	DRIVE_DS	3-00000086	I*4	DRIVE_DT
3-00000076	I*4	DRIVE_ER	3-00000082	I*4	DRIVE_FC
2-000008B0	I*4	DRIVE_FUNC	3-0000007A	I*4	DRIVE_MR
3-0000008E	I*4	DRIVE_SN	3-00000092	I*4	DRIVE_TC
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
2-000008A8	I*4	FIELD	2-000008B4	I*4	I
AP-00000004	L*1	LUN	2-000008AC	I*4	SELECTED_MAPPING_REGISTER

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000052	I*4	ADAPTER_REGISTERS	28	(7)
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)
2-00000086	CHAR	MASS1_CST	7	(0:0)
2-0000009D	CHAR	MASS1_DS	216	(0:8)
2-000003E7	CHAR	MASS1_DT	108	(0:3)
2-00000205	CHAR	MASS1_ER	180	(0:5)

```
  2-000004C7  CHAR MASS1_FORMAT                    39  (0:2)
  2-000004BB  CHAR MASS1_TC                        12  (3:3)
  2-0000008D  CHAR MASS2_CS1                       16  (11:11)
  2-00000175  CHAR MASS2_DS                       144  (10:15)
  2-00000471  CHAR MASS2_DT                        18  (10:10)
  2-0000031F  CHAR MASS2_ER                        36  (8:9)
  2-000004EE  CHAR MASS2_FORMAT                    39  (12:14)
  2-00000527  CHAR MASS2_TC                       128  (12:15)
  2-00000483  CHAR MASS3_DT                        10  (12:12)
  2-00000366  CHAR MASS3_ER                        96  (11:14)
  2-0000048D  CHAR MASS4_DT                        46  (14:15)
  2-00000515  CHAR MASS_DENSITY                    18  (3:4)
  2-00000453  CHAR MASS_DT5                        30  (0:1)
  2-00000034  CHAR MASS_ER10                       36  (0:1)
  2-00000058  CHAR MASS_ER15                       46  (0:1)
  2-00000000  CHAR MASS_ER7                        52  (0:1)
  2-000005A7  CHAR MASS_FUNC                      768  (0:31)
  2-000002EB  CHAR NRZI1_ER                        52  (6:7)
  2-00000354  CHAR NRZI2_ER                        18  (10:10)
  2-000003DD  CHAR NRZI3_ER                        10  (15:15)
  2-000002B9  CHAR PE1_ER                          50  (6:7)
  2-00000343  CHAR PE2_ER                          17  (10:10)
  2-000003C6  CHAR PE3_ER                          23  (15:15)
```

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-00000046	75'	1-00000059	100'	1-00000065	125'	1-00000076	140'	1-00000087	150'	1-00000098	160'
1-000000AF	175'	1-000000C0	183'	**	184	1-000000DC	185'	1-000000ED	200'	1-000000FE	215'
1-0000011C	225'	1-00000128	325'	1-00000139	330'	1-00000159	335'	**	340	1-0000017A	350'
1-00000188	375'	1-0000019C	400'	1-000001BC	425'	1-000001C8	450'	1-000001E0	500'	1-000001F8	530'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESSC		DHEAD1		FRCTOF
	IRP$L_PID		IRP$Q_IOSB		IRP$W_BCNT
	IRP$W_BOFF	I*4	LIB$EXTZV		LINCHR
	MASSTAPE_QIO		MBA_CONTROL_REGISTERS		MBA_MAPPING_REGISTER
	MBA_STATUS_REGISTER16_31		ORB$L_OWNER		OUTPUT
	UCB$B_ERTCNT		UCB$B_ERTMAX		UCB$L_CHAR
	UCB$L_OPCNT		UCB$W_ERRCNT		UCB$W_STS

```
0001
0002
0003
0004
0005          subroutine masstape_qio (lun,emb$w_dv_func)
0006
0007
0008
0009          include 'src$:qiocommon.for /nolist'
0273
0274
0275
0276          byte            lun
0277
0278          integer*2       emb$w_dv_func
0279
0280          integer*4       qiocode(0:1,0:63)
0281
0282
0283
0284
0285          if (qiocode(0,0) .eq. 0) then
0286
0287          qiocode(1,00) = %loc(io$_nop)
0288
0289          qiocode(1,01) = %loc(io$_unload)
0290
0291          qiocode(1,03) = %loc(io$_recal)
0292
0293          qiocode(1,04) = %loc(io$_drvclr)
0294
0295          qiocode(1,06) = %loc(io$_erasetape)
0296
0297          qiocode(1,08) = %loc(io$_packack)
0298
0299          qiocode(1,09) = %loc(io$_spacerecord)
0300
0301          qiocode(1,10) = %loc(io$_writecheck)
0302
0303          qiocode(1,11) = %loc(io$_writepblk)
0304
0305          qiocode(1,12) = %loc(io$_readpblk)
0306
0307          qiocode(1,25) = %loc(io$_readpreset)
0308
0309          qiocode(1,26) = %loc(io$_setchar)
0310
0311          qiocode(1,27) = %loc(io$_sensechar)
0312
0313          qiocode(1,28) = %loc(io$_writemark)
0314
0315          qiocode(1,32) = %loc(io$_writelblk)
0316
0317          qiocode(1,33) = %loc(io$_readlblk)
0318
0319          qiocode(1,34) = %loc(io$_rewindoff)
0320
```

```
0321            qiocode(1,35) = %loc(io$_setmode)
0322
0323            qiocode(1,36) = %loc(io$_rewind)
0324
0325            qiocode(1,37) = %loc(io$_skipfile)
0326
0327            qiocode(1,38) = %loc(io$_skiprecord)
0328
0329            qiocode(1,39) = %loc(io$_sensemode)
0330
0331            qiocode(1,40) = %loc(io$_writeof)
0332
0333            qiocode(1,48) = %loc(io$_writevblk)
0334
0335            qiocode(1,49) = %loc(io$_readvblk)
0336
0337            qiocode(1,50) = %loc(io$_access)
0338
0339            qiocode(1,51) = %loc(io$_create)
0340
0341            qiocode(1,52) = %loc(io$_deaccess)
0342
0343            qiocode(1,53) = %loc(io$_delete)
0344
0345            qiocode(1,54) = %loc(io$_modify)
0346
0347            qiocode(1,56) = %loc(io$_acpcontrol)
0348
0349            qiocode(1,57) = %loc(io$_mount)
0350
0351            do 10,i = 0,63
0352
0353            qiocode(0,i) = 33
0354
0355            if (qiocode(1,i) .eq. 0) then
0356
0357            qiocode(1,i) = %loc(qio_string)
0358            endif
0359
0360    10      continue
0361            endif
0362
0363            call irp$w_func (lun,emb$w_dv_func,
0364            1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))
0365
0366            return
0367
0368            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	311	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	548	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	QIOCOMMON	1247	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	2114	

ENTRY POINTS

Address	Type	Name
0-00000000		MASSTAPE_QIO

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008a	I*2	EMBSW_DV_FUNC	2-00000200	I*4	I
3-00000442	CHAR	IO$_ABORT	3-0000034D	CHAR	IO$_ACCESS
3-000003C2	CHAR	IO$_ACPCONTROL	3-000004B3	CHAR	IO$_AVAILABLE
3-00000297	CHAR	IO$_CLEAN	3-00000369	CHAR	IO$_CREATE
3-00000385	CHAR	IO$_DEACCESS	3-00000393	CHAR	IO$_DELETE
3-0000026D	CHAR	IO$_DIAGNOSE	3-00000065	CHAR	IO$_DRVCLR
3-000004CB	CHAR	IO$_DSE	3-000000A9	CHAR	IO$_ERASETAPE
3-00000276	CHAR	IO$_FORMAT	3-00000071	CHAR	IO$_INITIALIZE
3-00000014	CHAR	IO$_LOADMCODE	3-000003A1	CHAR	IO$_MODIFY
3-000003E2	CHAR	IO$_MOUNT	3-00000000	CHAR	IO$_NOP
3-0000009D	CHAR	IO$_OFFSET	3-000000EB	CHAR	IO$_PACKACK
3-000000E0	CHAR	IO$_QSTOP	3-000003EF	CHAR	IO$_RDSTATS
3-00000421	CHAR	IO$_READCSR	3-00000169	CHAR	IO$_READHEAD
3-000002B6	CHAR	IO$_READLBLK	3-0000013F	CHAR	IO$_READPBLK
3-00000200	CHAR	IO$_READPRESET	3-00000195	CHAR	IO$_READTRACKD
3-0000033A	CHAR	IO$_READVBLK	3-0000045A	CHAR	IO$_READWTHBUF
3-00000484	CHAR	IO$_READWTHXBUF	3-0000004D	CHAR	IO$_RECAL
3-0000007C	CHAR	IO$_RELEASE	3-000001AB	CHAR	IO$_REREADN
3-000001B8	CHAR	IO$_REREADP	3-000000CA	CHAR	IO$_RETCENTER
3-000002E6	CHAR	IO$_REWIND	3-000002C9	CHAR	IO$_REWINDOFF
3-000000FC	CHAR	IO$_SEARCH	3-00000024	CHAR	IO$_SEEK
3-00000231	CHAR	IO$_SENSECHAR	3-00000309	CHAR	IO$_SENSEMODE
3-0000021D	CHAR	IO$_SETCHAR	3-000003B8	CHAR	IO$_SETCLOCK
3-00000088	CHAR	IO$_SETCLOCKP	3-000002DD	CHAR	IO$_SETMODE
3-000002ED	CHAR	IO$_SKIPFILE	3-000002FA	CHAR	IO$_SKIPRECORD
3-00000029	CHAR	IO$_SPACEFILE	3-0000010E	CHAR	IO$_SPACERECORD
3-000003D7	CHAR	IO$_STARTDATA	3-000000B4	CHAR	IO$_STARTDATAP
3-00000037	CHAR	IO$_STARTMPROC	3-0000020F	CHAR	IO$_STARTSPNDL
3-00000059	CHAR	IO$_STOP	3-0000000D	CHAR	IO$_UNLOAD
3-0000046B	CHAR	IO$_WRITEBUFNCRC	3-0000011E	CHAR	IO$_WRITECHECK
3-000001E4	CHAR	IO$_WRITECHECKH	3-000003FF	CHAR	IO$_WRITECSR
3-00000153	CHAR	IO$_WRITEHEAD	3-000002A2	CHAR	IO$_WRITELBLK
3-00000247	CHAR	IO$_WRITEMARK	3-00000314	CHAR	IO$_WRITEOF
3-0000012A	CHAR	IO$_WRITEPBLK	3-000001C9	CHAR	IO$_WRITERET

```
  3-0000017E  CHAR IO$_WRITETRACKD          3-00000326  CHAR IO$_WRITEVBLK
  3-00000448  CHAR IO$_WRITEWTHBUF          3-00000257  CHAR IO$_WRTTMKR
 AP-0000004@  L*1  LUN                      3-000004A1  CHAR QIO_STRING
```

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	I*4	QIOCODE	512	(0:1, 0:63)

LABELS

Address	Label
**	10

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	IRP$W_FUNC	I*4	LIB$EXTZV

COMMAND QUALIFIERS

```
FORTRAN /LIS=LIS$:MASSTAPE/OBJ=OBJ$:MASSTAPE MSRC$:MASSTAPE

/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:           10.42 seconds
Elapsed Time:       21.63 seconds
Page Faults:        320
Dynamic Memory:     247 pages
```

0150 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LPA11K
LIS

MCHECKS
LIS

LCPRINTER
LIS

MBA
LIS

LOGMSCP
LIS

LINCHK
LIS

MBAINT
LIS

LABEL
LIS

LOGGER
LIS

MASSTAPE
LIS

MASSDISK
LIS