```
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
```

```
RRRRRRRR    LL          DDDDDDDD      IIIIII      SSSSSSSS  KK      KK
RRRRRRRR    LL          DDDDDDDD      IIIIII      SSSSSSSS  KK      KK
RR      RR  LL          DD      DD      II      SS          KK      KK
RR      RR  LL          DD      DD      II      SS          KK      KK
RR      RR  LL          DD      DD      II      SS          KK    KK
RR      RR  LL          DD      DD      II      SS          KK    KK
RRRRRRRR    LL          DD      DD      II        SSSSSS    KKKKKK
RRRRRRRR    LL          DD      DD      II        SSSSSS    KKKKKK
RR  RR      LL          DD      DD      II              SS  KK    KK
RR  RR      LL          DD      DD      II              SS  KK    KK
RR    RR    LL          DD      DD      II              SS  KK      KK      ....
RR    RR    LL          DD      DD      II              SS  KK      KK      ....
RR      RR  LLLLLLLLLL  DDDDDDDD      IIIIII    SSSSSSSS    KK      KK      ....
RR      RR  LLLLLLLLLL  DDDDDDDD      IIIIII    SSSSSSSS    KK      KK      ....


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 RLDISK (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       AUTHOR: BRIAN PORTER            CREATION DATE:5-FEB-1979
0030    C
0031
0032    C++
0033    C       Functional description:
0034    C
0035    C       This module displays entries made for the rl11 controller.
0036    C
0037    C       Modified by:
0038    C
0039    C       V03-004 SAR0232         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 SAR0093         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 SAR0050         Sharon A. Reynolds,     13-Jun-1983
0047    C               Removed brief/cryptic support.
0048    C
0049    C       v03-001 BP0001          Brian Porter,           20-AUG-1982
0050    C               Corrected device attention.
0051    C**
0052    C--
0053            INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
0112            INCLUDE 'SRC$:DEVERR.FOR /NOLIST'
0213
0214            INTEGER*4       RLCS
0215            INTEGER*4       RLBA
```

```
      INTEGER*4       RLDA
      INTEGER*4       RLMP
      INTEGER*4       UBA_REGS(4)
      INTEGER*4       FIELD
      INTEGER*4       FIELD1
      INTEGER*4       FIELD2
      INTEGER*4       COMPRESS4
      INTEGER*4       COMPRESSC

      BYTE            DRIVE_FUNC
      BYTE            LUN

      EQUIVALENCE     (RLCS,EMB$L_DV_REGSAV(0))
      EQUIVALENCE     (RLBA,EMB$L_DV_REGSAV(1))
      EQUIVALENCE     (RLDA,EMB$L_DV_REGSAV(2))
      EQUIVALENCE     (RLMP,EMB$L_DV_REGSAV(3))
      EQUIVALENCE     (UBA_REGS,EMB$L_DV_REGSAV(4))

      PARAMETER       NO_OPERATION    = 0
      PARAMETER       XFER_ERR        = 1
      PARAMETER       WRITE_CHECK     = 1
      PARAMETER       GET_STATUS      = 2
      PARAMETER       SEEK            = 3
      PARAMETER       READ_HEADER     = 4
      PARAMETER       WRITE_DATA      = 5
      PARAMETER       READ_DATA       = 6
      PARAMETER       TIMEOUT         = 96

      CHARACTER*12    V1RL_CS(0:0)
      DATA    V1RL_CS(0)      /'DRIVE READY*'/

      CHARACTER*17    V2RL_CS(6:7)
      DATA    V2RL_CS(6)      /'INTERRUPT ENABLE*'/
      DATA    V2RL_CS(7)      /'CONTROLLER READY*'/

      CHARACTER*21    V3RL_CS(10:10)
      DATA    V3RL_CS(10)     /'OPERATION INCOMPLETE*'/

      CHARACTER*20    V4RL_CS(13:15)
      DATA    V4RL_CS(13)     /'NON-EXISTENT MEMORY*'/
      DATA    V4RL_CS(14)     /'DRIVE ERROR*'/
      DATA    V4RL_CS(15)     /'COMPOSITE ERROR*'/

      CHARACTER*17    DCRC_HCRC(0:1)
      DATA    DCRC_HCRC(0)    /'DATA CRC ERROR*'/
      DATA    DCRC_HCRC(1)    /'HEADER CRC ERROR*'/

      CHARACTER*22    DLT_HNF(0:1)
      DATA    DLT_HNF(0)      /'DATA LATE ERROR*'/
      DATA    DLT_HNF(1)      /'HEADER NOT FOUND*'/

      CHARACTER*6     V1RL_DA(3:3)
      DATA    V1RL_DA(3)      /'RESET*'/

      CHARACTER*7     DIRECTION(0:1)
      DATA    DIRECTION(0)    /'REVERSE'/
      DATA    DIRECTION(1)    /'FORWARD'/
```

```
0273
0274            CHARACTER*24    HEAD_SELECT(0:1)
0275            DATA    HEAD_SELECT(0)  /'HEAD SELECT = UPPER HEAD'/
0276            DATA    HEAD_SELECT(1)  /'HEAD SELECT = LOWER HEAD'/
0277
0278            CHARACTER*2     ADDR_EXTN(4:5)
0279            DATA    ADDR_EXTN(4)    /'16'/
0280            DATA    ADDR_EXTN(5)    /'17'/
0281
0282            CHARACTER*13    V2RL_STATUS(3:5)
0283            DATA    V2RL_STATUS(3)  /'BRUSHES HOME*'/
0284            DATA    V2RL_STATUS(4)  /'HEADS OUT*'/
0285            DATA    V2RL_STATUS(5)  /'COVER OPEN*'/
0286
0287            CHARACTER*19    V3RL_STATUS(8:15)
0288            DATA    V3RL_STATUS(8)  /'DRIVE SELECT ERROR*'/
0289            DATA    V3RL_STATUS(9)  /'VOLUME CHECK*'/
0290            DATA    V3RL_STATUS(10) /'WRITE GATE ERROR*'/
0291            DATA    V3RL_STATUS(11) /'SPIN ERROR*'/
0292            DATA    V3RL_STATUS(12) /'SEEK TIME-OUT*'/
0293            DATA    V3RL_STATUS(13) /'WRITE PROTECTED*'/
0294            DATA    V3RL_STATUS(14) /'HEAD CURRENT ERROR*'/
0295            DATA    V3RL_STATUS(15) /'WRITE DATA ERROR*'/
0296
0297            CHARACTER*4     DRIVE_TYPE(0:1)
0298            DATA    DRIVE_TYPE(0)   /'RL01'/
0299            DATA    DRIVE_TYPE(1)   /'RL02'/
0300
0301            CHARACTER*15    V1RL_STATUS(0:7)
0302            DATA    V1RL_STATUS(0)  /'LOAD CARTRIDGE*'/
0303            DATA    V1RL_STATUS(1)  /'SPIN UP*'/
0304            DATA    V1RL_STATUS(2)  /'BRUSH CYCLE*'/
0305            DATA    V1RL_STATUS(3)  /'LOAD HEADS*'/
0306            DATA    V1RL_STATUS(4)  /'SEEK*'/
0307            DATA    V1RL_STATUS(5)  /'LOCK ON*'/
0308            DATA    V1RL_STATUS(6)  /'UNLOAD HEADS*'/
0309            DATA    V1RL_STATUS(7)  /'SPIN DOWN*'/
0310
0311            CHARACTER*23    RL_FUNC(0:7)
0312            DATA    RL_FUNC(0)      /'NO OPERATION*'/
0313            DATA    RL_FUNC(1)      /'WRITE CHECK*'/
0314            DATA    RL_FUNC(2)      /'GET STATUS*'/
0315            DATA    RL_FUNC(3)      /'SEEK*'/
0316            DATA    RL_FUNC(4)      /'READ HEADER*'/
0317            DATA    RL_FUNC(5)      /'WRITE DATA*'/
0318            DATA    RL_FUNC(6)      /'READ DATA*'/
0319            DATA    RL_FUNC(7)      /'READ DATA NO HDR CHECK*'/
0320
0321            CHARACTER*33    MUST_BE
0322            DATA    MUST_BE /'MUST BE SET/CLEAR INCORRECT'/
0323
0324
0325            CALL FRCTOF (LUN)
0326
0327            call dhead1 (lun,'UBA RL11')
0328
0329            CALL LINCHK (LUN,2)
```

```
0330
0331            WRITE(LUN,100) RLCS
0332    100     FORMAT(/' ',T8,'RLCS',T24,Z8.4)
0333
0334            CALL OUTPUT (LUN,RLCS,V1RL_CS,0,0,0,'0')
0335
0336            DRIVE_FUNC = LIB$EXTZV(1,3,RLCS)
0337
0338            CALL LINCHK (LUN,1)
0339
0340            WRITE(LUN,125) RL_FUNC(DRIVE_FUNC)
0341    125     FORMAT(' ',T40,'FONCTION = '
0342            1 A<COMPRESSC (RL_FUNC(DRIVE_FUNC))>)
0343
0344            DO 140,I = 4,5
0345
0346            FIELD - LIB$EXTZV(I,1,RLCS)
0347
0348            IF (FIELD .NE. 0) THEN
0349
0350            CALL LINCHK (LUN,1)
0351
0352            WRITE(LUN,130) ADDR_EXTN(I)
0353    130     FORMAT(' ',T40,'BUS ADDRESS EXTENSION BIT ',A2,'.')
0354            ENDIF
0355
0356    140     CONTINUE
0357
0358            CALL OUTPUT (LUN,RLCS,V2RL_CS,6,6,7,'0')
0359
0360            CALL LINCHK (LUN,1)
0361
0362            FIELD = LIB$EXTZV(8,2,RLCS)
0363
0364            WRITE(LUN,150) FIELD
0365    150     FORMAT(' ',T40,'SELECTED DRIVE = ',I1,'.')
0366
0367            CALL OUTPUT (LUN,RLCS,V3RL_CS,10,10,10,'0')
0368
0369            FIELD = LIB$EXTZV(10,1,RLCS)
0370
0371            FIELD1 = LIB$EXTZV(11,1,RLCS)
0372
0373            IF (FIELD1 .EQ. 1) THEN
0374
0375            CALL LINCHK (LUN,1)
0376
0377            IF (DRIVE_FUNC .NE. WRITE_CHECK) THEN
0378
0379            WRITE(LUN,160) DCRC_HCRC(FIELD)
0380    160     FORMAT(' ',T40,A<COMPRESSC (DCRC_HCRC(FIELD))>)
0381            ELSE
0382
0383            WRITE(LUN,170)
0384    170     FORMAT(' ',T40,'WRITE CHECK ERROR')
0385            ENDIF
0386            ENDIF
```

```
            FIELD1 = LIB$EXTZV(12,1,RLCS)

            IF (FIELD1 .NE. 0) THEN

            CALL LINCHK (LUN,1)

            WRITE(LUN,180) DLT_HNF(FIELD)
    180     FORMAT(' ',T40,A<INDEX(DLT_HNF(FIELD),'*')-1>)
            ENDIF

            CALL OUTPUT (LUN,RLCS,V4RL_CS,13,13,15,'0')

            CALL LINCHK (LUN,1)

            WRITE(LUN,250) RLBA
    250     FORMAT(' ',T8,'RLBA',T24,Z8.4)

            IF (DRIVE_FUNC .EQ. WRITE_CHECK
            1 .OR.
            2 DRIVE_FUNC .GE. WRITE_DATA) THEN

            CALL CALC_MAP (LUN,4,RLCS,RLBA)
            ENDIF

            CALL LINCHK (LUN,1)

            WRITE(LUN,275) RLDA
    275     FORMAT(' ',T8,'RLDA',T24,Z8.4)

            IF (DRIVE_FUNC .EQ. SEEK) THEN

            IF (JIAND(RLDA,'A'X) .NE. 'A'X
            1 .OR.
            2 JIAND(RLDA,'1'X) .EQ. 0) THEN

            CALL LINCHK (LUN,1)

            WRITE(LUN,280) MUST_BE
    280     FORMAT(' ',T40,A33)
            ENDIF

            FIELD = LIB$EXTZV(2,1,RLDA)

            CALL LINCHK (LUN,1)

            WRITE(LUN,285) DIRECTION(FIELD)
    285     FORMAT(' ',T40,A7)

            FIELD = LIB$EXTZV(4,1,RLDA)

            CALL LINCHK (LUN,1)

            WRITE(LUN,290) HEAD_SELECT(FIELD)
    290     FORMAT(' ',T40,A24)

            FIELD = LIB$EXTZV(7,9,RLDA)
```

```
0444
0445            CALL LINCHK (LUN,1)
0446
0447            WRITE(LUN,295) FIELD
0448    295     FORMAT(' ',T40,'CYLINDER DIFFERENCE = ',I<COMPRESS4 (FIELD)>,'.')
0449
0450            ELSE IF (DRIVE_FUNC .EQ. GET_STATUS) THEN
0451
0452            IF (JIAND(RLDA,'F4'X) .NE. 0
0453            1 .OR.
0454            2 JIAND(RLDA,'3'X) .NE. '3'X) THEN
0455
0456            CALL LINCHK (LUN,1)
0457
0458            WRITE(LUN,300) MUST_BE
0459    300     FORMAT(' ',T40,A33)
0460            ENDIF
0461
0462            CALL OUTPUT (LUN,RLDA,V1RL_DA,3,3,3,'0')
0463
0464            ELSE IF (DRIVE_FUNC .EQ. WRITE_CHECK
0465            1 .OR.
0466            2 DRIVE_FUNC .GE. WRITE_DATA) THEN
0467
0468            FIELD = LIB$EXTZV(0,6,RLDA)
0469
0470            CALL LINCHK (LUN,1)
0471
0472            WRITE(LUN,305) FIELD
0473    305     FORMAT(' ',T40,'SECTOR = ',I<COMPRESS4 (FIELD)>,'.')
0474
0475            FIELD = LIB$EXTZV(6,1,RLDA)
0476
0477            CALL LINCHK (LUN,1)
0478
0479            WRITE(LUN,310) HEAD_SELECT(FIELD)
0480    310     FORMAT(' ',T40,A24)
0481
0482            FIELD = LIB$EXTZV(7,9,RLDA)
0483
0484            CALL LINCHK (LUN,1)
0485
0486            WRITE(LUN,315) FIELD
0487    315     FORMAT(' ',T40,'CYLINDER = ',I<COMPRESS4 (FIELD)>,'.')
0488            ENDIF
0489
0490    C
0491    C       THE RLMP REGISTER RETURNED TO SYE IS NOT THE CONTENTS AT THE
0492    C       TIME OF THE ERROR.IT IS THE CONTENTS AFTER A GET STATUS
0493    C       COMMAND HAS BEEN EXECUTED ON THE FAILING DRIVE.A GET STATUS
0494    C       IS EXECUTED UNCONDITIONALLY WHEN ANY ERROR IS DETECTED BY
0495    C       THE RL DISK DRIVER.
0496    C
0497
0498            CALL LINCHK (LUN,2)
0499
0500            WRITE(LUN,605) RLMP
```

```
605     FORMAT(' ',T8,'RLMP',T24,Z8.4,/,T48,'**ERROR STATUS**')

        FIELD = LIB$EXTZV(0,3,RLMP)

        CALL LINCHK (LUN,1)

        WRITE(LUN,615) V1RL_STATUS(FIELD)
615     FORMAT(' ',T40,A<INDEX(V1RL_STATUS(FIELD),'*')-1>)

        CALL OUTPUT (LUN,RLMP,V2RL_STATUS,3,3,5,'0')

        FIELD = LIB$EXTZV(6,1,RLMP)

        CALL LINCHK (LUN,1)

        WRITE(LUN,620) HEAD_SELECT(FIELD)
620     FORMAT(' ',T40,A24)

        FIELD = LIB$EXTZV(7,1,RLMP)

        CALL LINCHK (LUN,1)

        WRITE(LUN,625) DRIVE_TYPE(FIELD)
625     FORMAT(' ',T40,'DRIVE IS ',A4)

        CALL OUTPUT (LUN,RLMP,V3RL_STATUS,8,8,15,'0')

        if (emb$w_hd_entry .ne. 98) then

        IF ((DRIVE_FUNC .EQ. WRITE_CHECK
        1 .OR.
        2 DRIVE_FUNC .GE. WRITE_DATA)
        3 .AND.
        4 EMB$W_HD_ENTRY .NE. TIMEOUT) THEN

        if (uba_regs(1) .ne. 0) then

        call uba_datapath (lun,uba_regs(1),uba_regs(2))
        endif

        call calc_map2 (4,rlcs,rlba,field)

        call uba_mapping (lun,field,uba_regs(3))

        if (
        1 lib$extzv(16,16,emb$l_dv_iosb1) .gt. 512
        1 .and.
        1 field .ne. 0
        1 ) then

        call uba_mapping (lun,(field-1),uba_regs(4))
        endif
        ENDIF
        endif

        call linchk (lun,1)
```

```
      write(lun,630)
630   format(' ',:)

      if (emb$w_hd_entry .ne. 98) then

      call ucb$b_ertcnt (lun,emb$b_dv_ertcnt)

      call ucb$b_ertmax (lun,emb$b_dv_ertmax)
      endif

      call orb$l_owner (lun,emb$l_dv_ownuic)

      call ucb$l_char (lun,emb$l_dv_char)

      call ucb$w_sts (lun,emb$w_dv_sts)

      call ucb$l_opcnt (lun,emb$l_dv_opcnt)

      call ucb$w_errcnt (lun,emb$w_dv_errcnt)

      if (emb$w_hd_entry .ne. 98) then

      call ucb$l_media (lun,emb$l_dv_media)

      call linchk (lun,1)

      write(lun,630)

      call rldisk_qio (lun,emb$w_dv_func)

      call irp$w_bcnt (lun,emb$w_dv_bcnt)

      call irp$w_boff (lun,emb$w_dv_boff)

      call irp$l_pid (lun,emb$l_dv_rqpid)

      call irp$q_iosb (lun,emb$l_dv_iosb1)
      endif

      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	2260	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	487	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	1928	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	5187	

ENTRY POINTS

Address	Type	Name
0-00000000		RLDISK

VARIABLES

Address	Type	Name	Address	Type	Name
2-0000030C	L*1	DRIVE_FUNC	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	2-00000330	I*4	FIELD
2-00000334	I*4	FIELDT	2-00000338	I*4	FIELD2
2-0000033C	I*4	I	AP-00000004@	L*1	LUN
2-0000030D	CHAR	MUST_BE	3-00000056	I*4	RLBA
3-00000052	I*4	RLCS	3-0000005A	I*4	RLDA
3-0000005E	I*4	RLMP			

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000111	CHAR	ADDR_EXTN	4	(4:5)
2-0000007F	CHAR	DCRC_HCRC	34	(0:1)
2-000000D3	CHAR	DIRECTION	14	(0:1)
2-000000A1	CHAR	DLT_HNF	44	(0:1)
2-000001D4	CHAR	DRIVE_TYPE	8	(0:1)
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)
2-000000E1	CHAR	HEAD_SELECT	48	(0:1)
2-00000254	CHAR	RL_FUNC	184	(0:7)

3-00000062	I*4	UBA_REGS	16	(4)
2-00000000	CHAR	V1RL_CS	12	(0:0)
2-000000CD	CHAR	V1RL_DA	6	(3:3)
2-000001DC	CHAR	V1RL_STATUS	120	(0:7)
2-0000000C	CHAR	V2RL_CS	34	(6:7)
2-00000115	CHAR	V2RL_STATUS	39	(3:5)
2-0000002E	CHAR	V3RL_CS	21	(10:10)
2-0000013C	CHAR	V3RL_STATUS	152	(8:15)
2-00000043	CHAR	V4RL_CS	60	(13:15)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-0000004D	100'	1-0000005F	125'	1-00000078	130'	**	140	1-0000009F	150'	1-000000BD	160'
1-000000C9	170'	1-000000E2	180'	1-000000EE	250'	1-000000FF	275'	1-00000110	280'	1-00000118	285'
1-00000120	290'	1-00000128	295'	1-0000014F	300'	1-00000157	305'	1-00000171	310'	1-00000179	315'
1-00000195	605'	1-000001BB	615'	1-000001C7	620'	1-000001CF	625'	1-000001E2	630'		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
	CALC_MAP		CALC_MAP2	I*4	COMPRESS4	I*4	COMPRESSC		DHEAD1		FRCTOF
	IRP$C_PID		IRP$Q_IOSB		IRP$W_BCNT		IRP$W_BOFF	I*4	LIB$EXTZV	I*4	LIB$INDEX
	LINCHR		ORB$L_OWNER		OUTPUT		RLDISK_QIO		UBA_DATAPATH		UBA_MAPPING
	UCB$B_ERTCNT		UCB$B_ERTMAX		UCB$L_CHAR		UCB$L_MEDIA		UCB$L_OPCNT		UCB$W_ERRCNT
	UCB$W_STS										

```
      Subroutine RLDISK_QIO (lun,emb$w_dv_func)
      include 'src$:qiocommon.for /nolist'
      byte            lun
      integer*2       emb$w_dv_func
      integer*4       qiocode(0:1,0:63)

      if (qiocode(0,0) .eq. 0) then
      qiocode(1,00) = %loc(io$_nop)
      qiocode(1,02) = %loc(io$_seek)
      qiocode(1,04) = %loc(io$_drvclr)
      qiocode(1,08) = %loc(io$_packack)
      qiocode(1,10) = %loc(io$_writecheck)
      qiocode(1,11) = %loc(io$_writepblk)
      qiocode(1,12) = %loc(io$_readpblk)
      qiocode(1,14) = %loc(io$_readhead)
      qiocode(1,26) = %loc(io$_setchar)
      qiocode(1,27) = %loc(io$_sensechar)
      qiocode(1,32) = %loc(io$_writelblk)
      qiocode(1,33) = %loc(io$_readlblk)
      qiocode(1,35) = %loc(io$_setmode)
      qiocode(1,39) = %loc(io$_sensemode)
      qiocode(1,48) = %loc(io$_writevblk)
      qiocode(1,49) = %loc(io$_readvblk)
      qiocode(1,50) = %loc(io$_access)
      qiocode(1,51) = %loc(io$_create)
      qiocode(1,52) = %loc(io$_deaccess)
      qiocode(1,53) = %loc(io$_delete)
      qiocode(1,54) = %loc(io$_modify)
```

```
0321
0322            qiocode(1,56) = %loc(io$_acpcontrol)
0323
0324            qiocode(1,57) = %loc(io$_mount)
0325
0326            do 10,i = 0,63
0327
0328            qiocode(0,i) = 33
0329
0330            if (qiocode(1,i) .eq. 0) then
0331
0332            qiocode(1,i) = %loc(qio_string)
0333            endif
0334
0335    10      continue
0336            endif
0337
0338            call irp$w_func (lun,emb$w_dv_func,
0339            1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))
0340
0341            return
0342
0343            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	249	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	2052	

ENTRY POINTS

Address	Type	Name
0-00000000		RLDISK_QIO

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008a	I*2	EMB$W_DV_FUNC	2-00000200	I*4	I
3-00000442	CHAR	IO$_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-00000004@ 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$:RLDISK/OBJ=OBJ$:RLDISK MSRC$:RLDISK

  /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:            9.03 seconds
  Elapsed Time:        19.09 seconds
  Page Faults:         248
  Dynamic Memory:      228 pages
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RECSELECT
LIS

RLDISK
LIS

PUDRIVER
LIS

PCL11T
LIS

ROLLUP
LIS

RXDISK
LIS

PCL11R
LIS

SB11
LIS

RKDISK
LIS