```
(RPAQQ DISKDLIONCOMS
       ((FNS \DL.DISKINIT \DL.DOBNOCROSSP \DL.GETDISKADDRESS \DL.HANDLEMULTIRUN \DL.INIT.DOB.CHAIN
             \DL.OBTAINNEWDOB \DL.PUTDISKADDRESS \DL.RECALIBRATE \DL.RELEASEDOB \DL.SHUGART.XFERDISK
             \DL.TRANSFERPAGE \DLDISK.EXECUTE \DL.ACTONVMEMFILE \DL.ACTONVMEMPAGE \DL.DISKSEEK \DL.XFERDISK
             \DL.DISKOP \DL.TRI.INITIATE.IO \DL.TRI.XFERDISK)
        (FNS \DISKDLION.INIT)
        (DECLARE%: DONTEVAL@LOAD DOCOPY (INITVARS \DLDISKSHAPE.SECTORSPERCYLINDER \DLDISKSHAPE.HEADSPERDRIVE
                                                 \DLDISKSHAPE.SECTORSPERHEAD \DL.FAULTHANDLERDOB \DL.DOBPAGE
                                                 \DL.DOBCHAIN \DL.SPAREDOB \DISKTYPE \CONTROLLERTYPE)
             (P (\DISKDLION.INIT)))
        (DECLARE%: EVAL@COMPILE DONTCOPY (COMS * DISKDLDECLS)
             (COMS * TRIDISKDLDECLS))))

(DEFINEQ

(\DL.DISKINIT
  [LAMBDA NIL                                                      ; Edited 30-Nov-87 15:06 by amd
    (DECLARE (GLOBALVARS \DLDISKSHAPE.SECTORSPERHEAD \DLDISKSHAPE.HEADSPERDRIVE \DLDISKSHAPE.SECTORSPERCYLINDER
                         \DISKTYPE \CONTROLLERTYPE \DL.SPAREDOB \DL.FAULTHANDLERDOB))

    ;; Determine controller type by looking on IOCB page for the password.  yes, I know this is crude, but until I fix both initials, it'll have to do.

    [COND
      [(EQ \DL.TRI.PASSWORD (fetch (TRIDENTIOCB TH.HEADER.OP) of \IOCBPAGE))

       ;; If the password matches, we have a Trident (or SMD) disk.  Try to figure out which disk it is.

       (SETQ \CONTROLLERTYPE '\TRIDENT)

       ;; Look at some remnants of trident initial to see if we're on a t80 / t300

       (COND
         ((NEQ 19 (fetch (TRIDENTIOCB TH.HDSFROMINITIAL) of \IOCBPAGE))
          (SETQ \DLDISKSHAPE.SECTORSPERHEAD 30)
          (SETQ \DLDISKSHAPE.HEADSPERDRIVE 5)
          (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 150)
          (SETQ \DISKTYPE '\T80))
         (T (SETQ \DLDISKSHAPE.SECTORSPERHEAD 30)
            (SETQ \DLDISKSHAPE.HEADSPERDRIVE 19)
            (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 570)
            (SETQ \DISKTYPE '\T300]
      (T  ;; Determines disk shape.  \DEVICE.INPUT 3 is the KStatus register, \DEVICE.INPUT 6 is the KTest register.  WARNING:  Both of these
          ;; registers read as their complements!

       (SETQ \CONTROLLERTYPE '\SHUGART)

       ;; First, read work 15 of the io page. This is set to 10/11 resp. for an M2242/M2243 on the Kiku.

       (COND
         ((EQ (\GETBASE \IOPAGE 15)
              10)                                                  ; M2242B: 50Mb Shugart-type disk.
          (SETQ \DISKTYPE '\M2242)
          (SETQ \DLDISKSHAPE.SECTORSPERHEAD 16)
          (SETQ \DLDISKSHAPE.HEADSPERDRIVE 7)
          (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 112))
         ((EQ (\GETBASE \IOPAGE 15)
              11)                                                  ; M2243B: 80Mb Shugart-type disk.
          (SETQ \DISKTYPE '\M2243)
          (SETQ \DLDISKSHAPE.SECTORSPERHEAD 16)
          (SETQ \DLDISKSHAPE.HEADSPERDRIVE 11)
          (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 176))
         ((NEQ 0 (LOGAND 32 (\DEVICE.INPUT 3)))                    ; An SA4000 is easy.  The SA4000 bit is set inb KStatus
          (SETQ \DISKTYPE '\SA4000)
          (SETQ \DLDISKSHAPE.SECTORSPERHEAD 28)
          (SETQ \DLDISKSHAPE.HEADSPERDRIVE 8)
          (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 224))
         (T  ;; Turn on HeadSelect16 in KCtl, which is tied to Sector' for Q2080s

          (\DEVICE.OUTPUT 32768 3)
          (COND
            ((EQ 0 (LOGAND 64 (\DEVICE.INPUT 6)))                  ; KStatus.notsector was low when we set HeadSelect16 high, so
                                                                   ; must be the old shugart SA1000

             (SETQQ \DISKTYPE \SA1000)
             (SETQ \DLDISKSHAPE.SECTORSPERHEAD 16)
             (SETQ \DLDISKSHAPE.HEADSPERDRIVE 4)
```

```
                        (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 64))
                    (T                                                      ; Ktest.Sector' is false, so it must be a regular quantum.  If it was
                                                                            ; a Q2080, bringing HS16 up would have brought Sector up too

                      ;; Turn off the head select bit and see if Ktest.sector' changed

                      (\DEVICE.OUTPUT 0 3)
                      (COND
                          ((NEQ 0 (LOGAND 64 (\DEVICE.INPUT 6)))   ; KStatus.notsector did not change.  Must be a Q2040 with
                                                                   ; NotSector tied high

                            (SETQQ \DISKTYPE \Q2040)
                            (SETQ \DLDISKSHAPE.SECTORSPERHEAD 16)
                            (SETQ \DLDISKSHAPE.HEADSPERDRIVE 8)
                            (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 128))
                          (T                                       ; KStatus.notsector did change, so must be Q2080 with
                                                                   ; NotSector tied to HeadSelect16

                            (SETQQ \DISKTYPE \Q2080)
                            (SETQ \DLDISKSHAPE.SECTORSPERHEAD 16)
                            (SETQ \DLDISKSHAPE.HEADSPERDRIVE 7)
                            (SETQ \DLDISKSHAPE.SECTORSPERCYLINDER 112]
        ;; Initialize the DOB chain

        (\DL.INIT.DOB.CHAIN)
        (SETQ \DL.FAULTHANDLERDOB (\DL.OBTAINNEWDOB T))
        (replace (DLION.DOB FAULTFLG) of (\DL.DOB.FROM.PAGE.OFFSET \DL.FAULTHANDLERDOB) with T)
        (SETQ \DL.SPAREDOB (\DL.OBTAINNEWDOB T))
        (replace (DLION.DOB FAULTFLG) of (\DL.DOB.FROM.PAGE.OFFSET \DL.SPAREDOB) with NIL])
```

(\DL.DOBNOCROSSP
```
  [LAMBDA (DOB)                                                             (* hts%: " 4-Jul-85 18:13")

            (* * Returns T if a DOB doesn't cross a cylinder boundary, NIL otherwise.
            Needs to normalize head and sector. Last page will be first page plus run length -
            1.0)

        (DECLARE (GLOBALVARS \DLDISKSHAPE.SECTORSPERCYLINDER \DLDISKSHAPE.SECTORSPERHEAD))
        (LET [(START.SECTOR (IPLUS (fetch (DLION.DOB SECTOR) of DOB)
                                   (ITIMES (fetch (DLION.DOB HEAD) of DOB)
                                           \DLDISKSHAPE.SECTORSPERHEAD]
            (EQ (IQUOTIENT START.SECTOR \DLDISKSHAPE.SECTORSPERCYLINDER)
                (IQUOTIENT (IPLUS START.SECTOR (SUB1 (OR (fetch (DLION.DOB RUNLENGTH) of DOB)
                                                         1)))
                           \DLDISKSHAPE.SECTORSPERCYLINDER]))
```

(\DL.GETDISKADDRESS
```
  [LAMBDA (DOB)                                                             (* mpl "27-Jun-85 10:18")
      (IPLUS (ITIMES (fetch (DLION.DOB CYLINDER) of DOB)
                     \DLDISKSHAPE.SECTORSPERCYLINDER)
             (ITIMES (fetch (DLION.DOB HEAD) of DOB)
                     \DLDISKSHAPE.SECTORSPERHEAD)
             (fetch (DLION.DOB SECTOR) of DOB])
```

(\DL.HANDLEMULTIRUN
```
  [LAMBDA (DOB)                                                             (* ejs%: "16-Sep-85 13:30")

          (* * For now, multi-page runs that cross cyl boundaries will be handled in a simple loop)

        (DECLARE (GLOBALVARS \DL.SPAREDOB \IOCBPAGE))
        (LET ((SPAREDOB (\DL.DOB.FROM.PAGE.OFFSET \DL.SPAREDOB)))
             (\BLT SPAREDOB DOB 16)
             (replace (DLION.DOB RUNLENGTH) of SPAREDOB with 1)
             (for I from 1 to (fetch (DLION.DOB RUNLENGTH) of DOB) bind STATUS
                do (SETQ STATUS (\DL.XFERDISK SPAREDOB))
                   (if (fetch (DLION.DOB INCDATAPTR) of DOB)
                       then (replace (DLION.DOB BUFFER) of SPAREDOB with (\ADDBASE (fetch (DLION.DOB BUFFER)
                                                                                         of SPAREDOB)
                                                                                  WORDSPERPAGE)))
                   (replace (DLION.DOB LABEL) of SPAREDOB with (LOCF (fetch (IOCBPAGE LID) of \IOCBPAGE)))
                   (replace (DLION.DOB SECTOR) of SPAREDOB with (ADD1 (fetch (DLION.DOB SECTOR) of SPAREDOB)))
                   (COND
                       ((NEQ STATUS 'OK)
                        (RETURN STATUS)))
                finally (RETURN STATUS])
```

(\DL.INIT.DOB.CHAIN
```
  [LAMBDA NIL                                                               (* ejs%: "14-Aug-85 23:46")

          (* * This fn creates the chain of DOBs that are used by the disk handler)

        (DECLARE (GLOBALVARS \DL.DOBPAGE \DL.DOBCHAIN)
               (CONSTANTS \DL.MAXDOBS \DL.DOBSIZE))
        (for I from 0 to \DL.MAXDOBS bind (CurrentAddr _ \DL.DOBPAGE) do (replace (DLION.DOB NEXTDOB) of CurrentAddr
                                                                            with (\ADDBASE CurrentAddr \DL.DOBSIZE))
                                                                         (replace (DLION.DOB STATUS) of CurrentAddr
```

```
                                                                with 'FREE)
                                                     (SETQ CurrentAddr (\ADDBASE CurrentAddr
                                                                        \DL.DOBSIZE))
        finally (replace (DLION.DOB NEXTDOB) of CurrentAddr with NIL)
                (replace (DLION.DOB STATUS) of CurrentAddr with 'FREE))
    (SETQ \DL.DOBCHAIN 0)]
```

(\DL.OBTAINNEWDOB
```
  [LAMBDA (AS.OFFSET.INTO.PAGE)                                        (* ejs%: "16-Sep-85 13:25")

            (* * Obtains a new DOB from the pool)

    (DECLARE (GLOBALVARS \DL.DOBCHAIN))
    (PROG (NEWONE TEMP)
          (COND
            [\DL.DOBCHAIN (SETQ NEWONE (\DL.DOB.FROM.PAGE.OFFSET \DL.DOBCHAIN))
                (SETQ TEMP (fetch (DLION.DOB NEXTDOB) of NEWONE))
                (COND
                  (TEMP (SETQ \DL.DOBCHAIN (\DL.PAGE.OFFSET.FROM.DOB TEMP)))
                  (T (SETQ \DL.DOBCHAIN NIL)))
                (with DLION.DOB NEWONE (SETQ MODE 'VRR)
                      (SETQ RUNLENGTH 1)
                      (SETQ BUFFER NIL)
                      (SETQ LABEL NIL)
                      (SETQ FAULTFLG NIL)
                      (SETQ INCDATAPTR NIL)
                      (SETQ UPDATESOURCELABEL NIL)
                      (SETQ STATUS 'INCOMPLETE)
                      (SETQ NEXTDOB NIL))
                (RETURN (COND
                          (AS.OFFSET.INTO.PAGE (\DL.PAGE.OFFSET.FROM.DOB NEWONE))
                          (T NEWONE]
            (T (RETURN NIL])
```

(\DL.PUTDISKADDRESS
```
  [LAMBDA (DOB DA)                                                     (* mpl "27-Jun-85 10:15")
    (DECLARE (GLOBALVARS \DLDISKSHAPE.SECTORSPERCYLINDER \DLDISKSHAPE.SECTORSPERHEAD))
    (replace (DLION.DOB CYLINDER) of DOB with (IQUOTIENT DA \DLDISKSHAPE.SECTORSPERCYLINDER))
    (replace (DLION.DOB HEAD) of DOB with (IQUOTIENT (IREMAINDER DA \DLDISKSHAPE.SECTORSPERCYLINDER)
                                                     \DLDISKSHAPE.SECTORSPERHEAD))
    (replace (DLION.DOB SECTOR) of DOB with (IREMAINDER (IREMAINDER DA \DLDISKSHAPE.SECTORSPERCYLINDER)
                                                        \DLDISKSHAPE.SECTORSPERHEAD])
```

(\DL.RECALIBRATE
```
  [LAMBDA NIL                                                          (* mpl "27-Jun-85 11:16")

            (* This function is used in the unlikely event that the disk drive fails to seek correctly.
            It will find track 0)

    (PROG ((PUNTCOUNTER 1100))
          (while (fetch (DLDISK.STATUS TRACK00) of (\DLDISK.GETSTATUS)) do [COND
                                                                  ((EQ (SETQ PUNTCOUNTER (SUB1
                                                                                           PUNTCOUNTER
                                                                                           ))
                                                                       0)
                                                                   (RAID 'CouldntRecalibrate]
                                                          (with IOCBPAGE \IOCBPAGE
                                                                (SETQ SSEEKCMD1 (LOGOR 128
                                                                                       \DL.MINUSSTEP
                                                                                       ))
                                                                (SETQ SSEEKCMD2 \DL.MINUSSTEP)
                                                                (SETQ SCYLINDERDISPLACEMENT
                                                                 MAX.SMALLP))
                                                          (\DL.DISKOP \DL.SEEKIOCBSTART)
                                                          (replace (IOCBPAGE HCYLINDER)
                                                             of \IOCBPAGE with 0])
```

(\DL.RELEASEDOB
```
  [LAMBDA (DOB)                                                        (* ejs%: "16-Sep-85 13:39")

            (* * Releases a DOB to the free pool)

    (DECLARE (GLOBALVARS \DL.DOBCHAIN))
    (replace (DLION.DOB STATUS) of DOB with 'FREE)
    (replace (DLION.DOB NEXTDOB) of DOB with (\DL.DOB.FROM.PAGE.OFFSET \DL.DOBCHAIN))
    (replace (DLION.DOB BUFFER) of DOB with NIL)
    (replace (DLION.DOB LABEL) of DOB with NIL)
    (SETQ \DL.DOBCHAIN (\DL.PAGE.OFFSET.FROM.DOB DOB))
    NIL])
```

(\DL.SHUGART.XFERDISK
```
  [LAMBDA (DOB)                                                        ; Edited 30-Nov-87 14:40 by amd
```

```
      (DECLARE (GLOBALVARS \DLDISKSHAPE.SECTORSPERHEAD \DLDISKSHAPE.HEADSPERDRIVE))
   (PROG ((IOCBPG \IOCBPAGE)
          (TriedRecalibrate NIL)
          (CYL (fetch (DLION.DOB CYLINDER) of DOB))
          (HD (fetch (DLION.DOB HEAD) of DOB))
          (SEC (fetch (DLION.DOB SECTOR) of DOB))
          (MODE (fetch (DLION.DOB MODE) of DOB))
          Status RetryCount)
```

;; Check for out of bounds triples and normalize

```
          (SETQ HD (IPLUS HD (IQUOTIENT SEC \DLDISKSHAPE.SECTORSPERHEAD)))
          (SETQ SEC (IREMAINDER SEC \DLDISKSHAPE.SECTORSPERHEAD))
          (SETQ CYL (IPLUS CYL (IQUOTIENT HD \DLDISKSHAPE.HEADSPERDRIVE)))
          (SETQ HD (IREMAINDER HD \DLDISKSHAPE.HEADSPERDRIVE))
```

;; First thing to do is seek the new track

```
    RETRY
          (SETQ RetryCount 10)
          (\DL.DISKSEEK CYL)
    LP
```

;; Compute and fill in the head and sector information for the IOCB's header field

```
          (replace (IOCBPAGE HHEAD) of IOCBPG with HD)
          (replace (IOCBPAGE HSECTOR) of IOCBPG with SEC)
```

;; We can do variable length runs now, but these runs MUST be in contiguous pages of virtual memory. Beware!

```
          [replace (IOCBPAGE PRUNLENGTH) of IOCBPG with (COND
                                                   ((EQ (fetch (DLION.DOB RUNLENGTH) of DOB)
                                                        0)
                                                    1)
                                                   (T (fetch (DLION.DOB RUNLENGTH) of DOB]
```

;; Now that we know what we're doing, dispatch on the mode and set up the rest of the parameters

```
          [with IOCBPAGE IOCBPG

               ;; Fill in IOCB fields according to mode

               (SETQ PDATALEN (COND
                                 ((fetch (DLION.DOB INCDATAPTR) of DOB)
                                  (LOGOR 256 32768))
                                 (T 256)))

               ;; Set the DATALEN variable to reflect wether we want to increment the page # or not

               (SELECTQ MODE
                   ((NIL VRR)
                        (SETQ PLABELLEN 12)
                        (SETQ PLABELCMD \DL.READFIELD)
                        (SETQ PLABELABORT \DL.ABORT-NR/WF/OVR/CRC)
                        (SETQ PDATACMD \DL.READFIELD)
                        (SETQ PDATAABORT \DL.ABORT-NR/WF/OVR/CRC))
                   (VVR (SETQ PLABELLEN 11)
                        (SETQ PLABELCMD \DL.VERIFYFIELD)
                        (SETQ PLABELABORT \DL.ABORT-NR/WF/OVR/CRC/VERIFY)
                        (SETQ PDATACMD \DL.READFIELD)
                        (SETQ PDATAABORT \DL.ABORT-NR/WF/OVR/CRC))
                   (VWW (SETQ PLABELLEN 12)
                        (SETQ PLABELCMD \DL.WRITEFIELD)
                        (SETQ PLABELABORT \DL.ABORT-NR/WF/OVR)
                        (SETQ PDATACMD \DL.WRITEFIELD)
                        (SETQ PDATAABORT \DL.ABORT-NR/WF/OVR))
                   (VVW (SETQ PLABELLEN 11)
                        (SETQ PLABELCMD \DL.VERIFYFIELD)
                        (SETQ PLABELABORT \DL.ABORT-NR/WF/OVR/CRC/VERIFY)
                        (SETQ PDATACMD \DL.WRITEFIELD)
                        (SETQ PDATAABORT \DL.ABORT-NR/WF/OVR/CRC))
                   ((T VRW)
                        (SETQ PLABELLEN 12)
                        (SETQ PLABELCMD \DL.READFIELD)
                        (SETQ PLABELABORT \DL.ABORT-NR/WF/OVR/CRC)
                        (SETQ PDATACMD \DL.WRITEFIELD)
                        (SETQ PDATAABORT \DL.ABORT-NR/WF/OVR))
                   (PROGN (RAID 'InvalidMode]
```

;; Fill in the virtual page field

;; If we're incrementing the page number, the microcode increments the page number FIRST before doing the transfer. We have to offset that by
;; subtracting one page

```
          [replace (IOCBPAGE PVPAGE) of IOCBPG with (COND
                                                     [(fetch (DLION.DOB INCDATAPTR) of DOB)
                                                      (SUB1 (fetch (POINTER PAGE#) of (fetch (DLION.DOB BUFFER)
                                                                                        of DOB]
                                                     (T (fetch (POINTER PAGE#) of (fetch (DLION.DOB BUFFER)
                                                                                    of DOB]
```

;; More important bit twiddling gibberish that I don't understand yet

```
          (replace (IOCBPAGE PTERMCOND1HEAD) of IOCBPG with HD)
          (replace (IOCBPAGE PTERMCOND2HEAD) of IOCBPG with HD)
```

;; If we were given a label, we better put it on the IOCB page

```
        (COND
            ((fetch (DLION.DOB LABEL) of DOB)
             (\BLT (LOCF (fetch (IOCBPAGE LID) of IOCBPG))
                   (fetch (DLION.DOB LABEL) of DOB)
                   10)))
```

;; Finally, the moment has come.. Wake up the microcode

```
        (SETQ Status (\DL.DISKOP \DL.XFERIOCBSTART))
```

;; For now, verify errors OK in the swapper. This is so we don't have to worry about boot links.

```
        (with IOCBPAGE IOCBPG (SETQ PLABELLEN 12)
             (SETQ PLABELCMD \DL.READFIELD)
             (SETQ PLABELABORT \DL.ABORT−NR/WF/OVR/CRC)
             (SETQ PDATACMD \DL.READFIELD)
             (SETQ PDATALEN (LOGOR 256 32768))
             (SETQ PDATAABORT \DL.ABORT−NR/WF/OVR/CRC))
```

;; Figure out what kind of error and update the field in the DOB

```
        [replace (DLION.DOB STATUS) of DOB with (SETQ Status (COND
                                                    ((OR (fetch (DLDISK.STATUS CRCERR) of Status)
                                                         (fetch (DLDISK.STATUS OVERRUN) of Status))
                                                     'READERROR)
                                                    ((fetch (DLDISK.STATUS WRITEFAULT) of Status)
                                                     'WRITEERROR)
                                                    ((fetch (DLDISK.STATUS VERIFYERR) of Status)
                                                     'VERIFYERROR)
                                                    (T 'OK]
```

;; If no error but verify error, then we can go about returning; otherwise, go on and try reading again.

```
        (COND
            ((OR (EQ Status 'OK)
                 (EQ Status 'VERIFYERROR))

             ;; See if we have to return a label and do so if need be.

             (COND
                ([AND (fetch (DLION.DOB LABEL) of DOB)
                      (OR (fetch (DLION.DOB UPDATESOURCELABEL) of DOB)
                          (EQ MODE 'VRR]
                 (\BLT (fetch (DLION.DOB LABEL) of DOB)
                       (LOCF (fetch (IOCBPAGE LID) of IOCBPG))
                       10)))

             ;; return the Status

             (RETURN Status)))
        (COND
            ((NEQ (SETQ RetryCount (SUB1 RetryCount))
                  0)
             (GO LP))
            (TriedRecalibrate

                    ;; We have already recalibrated and tried ten more times and failed, so we're really stuck

                    (RETURN Status))
            (T ;; Who knows?  We might be lost.  Recalibrate and try again

                (\DL.RECALIBRATE)
                (SETQ TriedRecalibrate T)
                (GO RETRY])
```

⟨\DL.TRANSFERPAGE

```
  [LAMBDA (DA BUFPTR OP LABPTR RUNSIZE NORAIDFLG)                    (* mpl "29-Jun-85 17:09")

        (* * This is only here as a stub till all callers go away!!!)

  (PROG (DOB STAT)
        (SETQ DOB (\DL.OBTAINNEWDOB))
        (COND
            ((NULL DOB)
             (RAID)))
        (with DLION.DOB DOB (SETQ DISKADDRESS DA)
            (SETQ BUFFER BUFPTR)
            (SETQ MODE OP)
            (SETQ LABEL LABPTR)
            (SETQ RUNLENGTH (OR RUNSIZE 1))
            (SETQ NEXTDOB NIL))
        (SETQ STAT (\DLDISK.EXECUTE DOB))
        (SETQ DOB (\DL.RELEASEDOB DOB))
        (RETURN (SELECTQ STAT
                    (OK T)
                    (WRITEERROR 8)
                    (READERROR 2)
                    (VERIFYERROR 1)
                    NIL])
```

⟨\DLDISK.EXECUTE

```
  [LAMBDA (DOBCHAIN)                                                 (* mpl " 3-Aug-85 18:55")
```

```
                        (* * Main call to the disk handler. Will process the chain of DOB's until an error is detected)

    (while (NEQ DOBCHAIN NIL) bind (LASTSTATUS _ 'OK)
        do                                                          (* Check to see if the run crosses a cylinder boundary)
            [COND
               [(OR (EQ \MACHINETYPE \DAYBREAK)
                    (EQ (OR (fetch (DLION.DOB RUNLENGTH) of DOBCHAIN)
                            1)
                        1)
                    (\DL.DOBNOCROSSP DOBCHAIN))
                (SETQ LASTSTATUS (SELECTC \MACHINETYPE
                                    (\DANDELION (\DL.XFERDISK DOBCHAIN))
                                    (\DAYBREAK (with DLION.DOB DOBCHAIN (\DOVE.XFERDISK CYLINDER HEAD SECTOR
                                                                            BUFFER MODE LABEL RUNLENGTH)))
                                    (PROGN (RAID]
               (T (SETQ LASTSTATUS (\DL.HANDLEMULTIRUN DOBCHAIN]
            (replace (DLION.DOB STATUS) of DOBCHAIN with LASTSTATUS)
            (SETQ DOBCHAIN (fetch (DLION.DOB NEXTDOB) of DOBCHAIN))
            (COND
               ((OR (NEQ LASTSTATUS 'OK)
                    (NULL DOBCHAIN))
                (RETURN LASTSTATUS])
```

DL.ACTONVMEMFILE
```
  [LAMBDA (FILEPAGE BUFFER NPAGES WRITEFLAG)                       (* bvm%: "12-Jun-85 16:32")
                                                                  (* Pilot page is zero-based, vmem page is one-base)

    (FRPTQ NPAGES (\DL.ACTONVMEMPAGE (SUB1 FILEPAGE)
                        BUFFER WRITEFLAG)
           (SETQ BUFFER (\ADDBASE BUFFER WORDSPERPAGE))
           (add FILEPAGE 1])
```

DL.ACTONVMEMPAGE
```
  [LAMBDA (FILEPAGE BUFPTR WRITEFLAG)                              (* bvm%: " 8-Jan-86 15:29")

            (* This fn brings in a page of the VMEM file. It scans over the bad page table, which contains a loose mapping of logical
            pages to physical pages.)

    (DECLARE (GLOBALVARS \DL.FAULTHANDLERDOB))
    (PROG ((LINKBASE (LOCF (fetch DLVMEMFILEINFO of \IOCBPAGE)))
           (I (IQUOTIENT (IDIFFERENCE (INDEXF (fetch FLOPPYIOCBSTART))
                                      (INDEXF (fetch DLVMEMFILEINFO)))
                         3))
           NEXTPAGE)
      LP  [COND
             ((OR (IGREATERP (SETQ NEXTPAGE (fetch DLFIRSTFILEPAGE of (fetch DLNEXTRUN of LINKBASE)))
                             FILEPAGE)
                  (EQ 0 NEXTPAGE))
              (with DLION.DOB (\DL.DOB.FROM.PAGE.OFFSET \DL.FAULTHANDLERDOB)
                    (SETQ CYLINDER (fetch DLVMCYL of LINKBASE))
                    (SETQ HEAD (fetch DLVMHEAD of LINKBASE))
                    (SETQ SECTOR (IPLUS (IDIFFERENCE FILEPAGE (fetch DLFIRSTFILEPAGE of LINKBASE))
                                        (fetch DLVMSECTOR of LINKBASE)))
                    (SETQ RUNLENGTH 1)
                    (SETQ BUFFER BUFPTR)
                    (SETQ LABEL NIL)
                    (SETQ MODE WRITEFLAG)
                    (SETQ NEXTDOB NIL)
                    (SETQ INCDATAPTR NIL)
                    (SETQ UPDATESOURCELABEL NIL))
              (RETURN (COND
                         ((NEQ (\DL.XFERDISK (\DL.DOB.FROM.PAGE.OFFSET \DL.FAULTHANDLERDOB))
                               'OK)
                          (\MP.ERROR \MP.SWAPDISKERROR "Hard Disk Error in swapper"]
             (COND
                ((EQ 0 I)
                 (\MP.ERROR \MP.BADRUNTABLE "Malformed run table for vmem file")
                 (RETURN)))
             (SETQ I (SUB1 I))
             (SETQ LINKBASE (fetch DLNEXTRUN of LINKBASE))
             (GO LP])
```

DL.DISKSEEK
```
  [LAMBDA (Cylinder)                                              (* bvm%: "12-Jun-85 16:37")

            (* Set DISPlacement to the difference between where we want to be and where we are.
            Negative DISP means step OUT in the positive direction)

    (PROG ((Displacement (IDIFFERENCE Cylinder (fetch (IOCBPAGE HCYLINDER) of \IOCBPAGE)))
           MicrocodeSeekCommand)                                  (* Don't try to move zero cylinders)
          (COND
             ((EQ Displacement 0)
              (RETURN 0)))
```

```
                    (* This magic constant is used by the microcode to form the step command.
                    \DL.MINUSSTEP is the reverse step and \DL.PLUSSTEP is the forward step.
                    You need this constant twice, but the second time it has a 128 OR'd into it.
                    as part of the step command microcode)

                    (SETQ MicrocodeSeekCommand \DL.PLUSSTEP)


                    (* The microcode always expects to see a negative number for the displacement, since it counts up to zero from it.
                    Thus, we have to make it negative if we are stepping to a cylinder that is farther out than we are now.)

                    (COND
                        ((ILESSP Displacement 0)
                         (SETQ Displacement (IMINUS Displacement)))
                        (T (SETQ MicrocodeSeekCommand \DL.MINUSSTEP)))          (* Deposit the seek commands onto the IOCB Page)
                    (replace (IOCBPAGE SSEEKCMD2) of \IOCBPAGE with MicrocodeSeekCommand)
                    (replace (IOCBPAGE SSEEKCMD1) of \IOCBPAGE with (LOGOR MicrocodeSeekCommand 128))
                                                                     (* Place the displacement value onto the IOCB Page)
                    (replace (IOCBPAGE SCYLINDERDISPLACEMENT) of \IOCBPAGE with (ADD1 (LOGXOR Displacement 65535)))
                                                                     (* Two's complement of Displacement --
                                                                     note it is not zero)
                                                                     (* Finally, we should update the header record to indicate our
                                                                     new position)
                    (replace (IOCBPAGE HCYLINDER) of \IOCBPAGE with Cylinder)


                    (* |...| and wake up the microcode to get things rolling! Note%: the 338 below is another DLion magic number, and it refers
                    to the distance into the I/O page + 256 where the beginning of the microprogram for the disk microcode is.
                    This is the beginning of the SEEK iocb. It is also used in \DL.RECALIBRATE, and the TRANSFER iocb is used in
                    \DL.XFERDISK)

                    (RETURN (\DL.DISKOP \DL.SEEKIOCBSTART])
```

(**DL.XFERDISK**
```
  [LAMBDA (DOB)                                                      ; Edited 30-Nov-87 15:40 by amd
    (DECLARE (GLOBALVARS \CONTROLLERTYPE))
    (COND
       [(EQ \CONTROLLERTYPE '\TRIDENT)                               ; We're using a Trident
        (replace (DLION.DOB STATUS) of DOB with (\DL.TRI.XFERDISK (fetch (DLION.DOB CYLINDER) of DOB)
                                                                   (fetch (DLION.DOB HEAD) of DOB)
                                                                   (fetch (DLION.DOB SECTOR) of DOB)
                                                                   (fetch (DLION.DOB BUFFER) of DOB)
                                                                   (fetch (DLION.DOB MODE) of DOB)
                                                                   (fetch (DLION.DOB LABEL) of DOB)
                                                                   (fetch (DLION.DOB RUNLENGTH) of DOB]
       (T                                                            ; We're using a Shugart
          (SELECTQ \DISKTYPE
             ((\SA4000 \M2242 \M2243)                                ; Bug somewhere that causes these drives to fail on VVW:
                                                                     ; change it to VRW to work around this.
                  (AND (EQ (FETCH (DLION.DOB MODE) OF DOB)
                           'VVW)
                       (REPLACE (DLION.DOB MODE) OF DOB WITH 'VRW)))
             NIL)
          (\DL.SHUGART.XFERDISK DOB])
```

(**DL.DISKOP**
```
  [LAMBDA (IOCB)                                                     (* bvm%: "12-Jun-85 13:15")


          (* Put something unreasonable onto the IOCB status so when it changes we will know it's over.)

    (replace (IOCBPAGE LASTIOCBSTATUS) of \IOCBPAGE with 256)        (* Start the microcode. There isn't a field in IOPAGE yet for this.)
    (\PUTBASE \IOPAGE 1 IOCB)                                        (* Issue a FirmWare Enable)
    (\DEVICE.OUTPUT 32 3)                                            (* wait for it to change back)
    (until (EQ (LOGAND (fetch (IOCBPAGE LASTIOCBSTATUS) of \IOCBPAGE)
                256)
            0))                                                      (* return status)
    (fetch (IOCBPAGE LASTIOCBSTATUS) of \IOCBPAGE])
```

(**DL.TRI.INITIATE.IO**
```
  [LAMBDA NIL                                                        (* bvm%: "12-Jun-85 13:14")
                                                                     (* This function starts the microcode for the Trident disk)
                                                                     (* Send wakeup to KCMD, register 13)
    (\DEVICE.OUTPUT \DL.TRI.STARTCMD 13)
    (until (EQ (LOGAND (fetch (TRIDENTIOCB TH.CONTROLLERSTATUS) of \IOCBPAGE)
                32768)
            0))
    (fetch (TRIDENTIOCB TH.CONTROLLERSTATUS) of \IOCBPAGE])
```

(**DL.TRI.XFERDISK**
```
  [LAMBDA (CYL HD SEC BUFFER MODE LABEL RUNLENGTH NORAIDFLG)         (* mpl " 2-Aug-85 15:51")


          (* This is \DL.TRI.XFERDISK, the lowest level driver for DLion Trident disks, capable of working with labels and retaining the
          same arguments the old one, but adding a few more%: NORAIDFLG LABEL, MODE, If USERFLG is T, errors will be
          returned, or else RAID is called. LABEL points to a LABEL record, Mode supplants WRITEFLG and can be VRR, VVR,
          VVW, or VWW.)
```

```
      (DECLARE (GLOBALVARS \DLDISKSHAPE.HEADSPERDRIVE \DLDISKSHAPE.SECTORSPERHEAD))
      (PROG ((RetryCount 10)
             Status USERFLG (FLG NIL)
             (Drive 0)
             DriveMask)                                              (* Check for out of bounds triples and normalize)
            (COND
                ((EQ RUNLENGTH 0)
                  (SETQ RUNLENGTH 1)))
            [SETQ DriveMask (LLSH 1 (IPLUS 8 (IDIFFERENCE 3 Drive]
            (SETQ HD (IPLUS HD (IQUOTIENT SEC \DLDISKSHAPE.SECTORSPERHEAD)))
            (SETQ SEC (IMOD SEC \DLDISKSHAPE.SECTORSPERHEAD))
            (SETQ CYL (IPLUS CYL (IQUOTIENT HD \DLDISKSHAPE.HEADSPERDRIVE)))
            (SETQ HD (IMOD HD \DLDISKSHAPE.HEADSPERDRIVE))           (* no recal first time)
            (replace (TRIDENTDCB RECALIBRATE) of \IOPAGE with NIL)
        LP                                                          (* Compute and fill in the head and sector information for the
                                                                       IOCB's header field)
            [with TRIDENTIOCB \IOCBPAGE (SETQ TH.CYL CYL)
                  (SETQ TH.HD HD)
                  (SETQ TH.SEC SEC)
                  (SETQ TH.CONTROLLERSTATUS 49152)
                  (SETQ TH.DISKSTATUS 0)
                  (SETQ TH.MICRONEXT 0)
                  (SETQ TH.DCBOFFSET 0)
                  [SETQ TH.LABELPTRLO (LOGAND 65280 (\LOLOC (LOCF (fetch (TRIDENTIOCB TH.READLABEL) of \IOPAGE]
                  (SETQ TH.LABELPTRHI (\HILOC (LOCF (fetch (TRIDENTIOCB TH.READLABEL) of \IOPAGE]
                                                                    (* Copy client header to uCode header)
            (with TRIDENTIOCB \IOCBPAGE (SETQ TH.uCYL TH.CYL)
                  (SETQ TH.uSEC TH.SEC)
                  (SETQ TH.uHD TH.HD))                              (* Set up the SEEK portion of the IOCB)
            (with TRIDENTIOCB \IOCBPAGE (SETQ TH.SEEK.XFERMASK 0)
                  (SETQ TH.SEEK.KDRIVE 63488)
                  (SETQ TH.SEEK.KCYL (LOGOR \TH.SEEK.CYLCMD CYL))
                  (SETQ TH.SEEK.KHEAD (LOGOR \TH.SEEK.HDCMD HD)))   (* Set up the HEADER portion of the IOCB)
            (with TRIDENTIOCB \IOCBPAGE (SETQ TH.HEADER.COMMAND \TH.HV.COMMAND)
                  (SETQ TH.HEADER.CONTROL \TH.HV.CONTROL)
                  (SETQ TH.HEADER.OP \TH.HV.OP)
                  (SETQ TH.HEADER.ERRORMASK \TH.HV.ERRORMASK))

      (* We can do variable length runs now, but these runs MUST be in contiguous pages of virtual memory.
      Beware!)

            (replace (TRIDENTIOCB TH.PAGECOUNT) of \IOCBPAGE with (COND
                                                                     (RUNLENGTH RUNLENGTH)
                                                                     (T 1)))

      (* Now that we know what we're doing, dispatch on the mode and set up the rest of the parameters)

            (SELECTQ MODE
                ((NIL VRR)
                    (with TRIDENTIOCB \IOCBPAGE (SETQ TH.LABEL.OP \TH.LR.OP)
                          (SETQ TH.LABEL.ERRORMASK \TH.LR.ERRORMASK)
                          (SETQ TH.LABEL.CONTROL (LOGOR \TH.LR.CONTROL DriveMask))
                          (SETQ TH.LABEL.COMMAND \TH.LR.COMMAND)
                          (SETQ TH.DATA.OP \TH.DR.OP)
                          (SETQ TH.DATA.ERRORMASK \TH.DR.ERRORMASK)
                          (SETQ TH.DATA.CONTROL (LOGOR \TH.DR.CONTROL DriveMask))
                          (SETQ TH.DATA.COMMAND \TH.DR.COMMAND)))
                (VVR (with TRIDENTIOCB \IOCBPAGE (SETQ TH.LABEL.OP \TH.LV.OP)
                          (SETQ TH.LABEL.ERRORMASK \TH.LV.ERRORMASK)
                          (SETQ TH.LABEL.CONTROL (LOGOR \TH.LV.CONTROL DriveMask))
                          (SETQ TH.LABEL.COMMAND \TH.LV.COMMAND)
                          (SETQ TH.DATA.OP \TH.DR.OP)
                          (SETQ TH.DATA.ERRORMASK \TH.DR.ERRORMASK)
                          (SETQ TH.DATA.CONTROL (LOGOR \TH.DR.CONTROL DriveMask))
                          (SETQ TH.DATA.COMMAND \TH.DR.COMMAND)))
                (VWW (with TRIDENTIOCB \IOCBPAGE (SETQ TH.LABEL.OP \TH.LW.OP)
                          (SETQ TH.LABEL.ERRORMASK \TH.LW.ERRORMASK)
                          (SETQ TH.LABEL.CONTROL (LOGOR \TH.LW.CONTROL DriveMask))
                          (SETQ TH.LABEL.COMMAND \TH.LW.COMMAND)
                          (SETQ TH.DATA.OP \TH.DW.OP)
                          (SETQ TH.DATA.ERRORMASK \TH.DW.ERRORMASK)
                          (SETQ TH.DATA.CONTROL (LOGOR \TH.WW.CONTROL DriveMask))
                          (SETQ TH.DATA.COMMAND \TH.DW.COMMAND)))
                (VVW (with TRIDENTIOCB \IOCBPAGE (SETQ TH.LABEL.OP \TH.LV.OP)
                          (SETQ TH.LABEL.ERRORMASK \TH.LV.ERRORMASK)
                          (SETQ TH.LABEL.CONTROL (LOGOR \TH.LV.CONTROL DriveMask))
                          (SETQ TH.DATA.OP \TH.DW.OP)
                          (SETQ TH.DATA.ERRORMASK \TH.DW.ERRORMASK)
                          (SETQ TH.DATA.CONTROL (LOGOR \TH.DW.CONTROL DriveMask))
                          (SETQ TH.DATA.COMMAND \TH.DW.COMMAND)))
                ((T VRW)
                    (with TRIDENTIOCB \IOCBPAGE (SETQ TH.LABEL.OP \TH.LR.OP)
                          (SETQ TH.LABEL.ERRORMASK \TH.LR.ERRORMASK)
                          (SETQ TH.LABEL.CONTROL (LOGOR \TH.LR.CONTROL DriveMask))
                          (SETQ TH.LABEL.COMMAND \TH.LR.COMMAND)
                          (SETQ TH.DATA.OP \TH.DW.OP)
```

```
                                (SETQ TH.DATA.ERRORMASK \TH.DW.ERRORMASK)
                                (SETQ TH.DATA.CONTROL (LOGOR \TH.DW.CONTROL DriveMask))
                                (SETQ TH.DATA.COMMAND \TH.DW.COMMAND)))
                    (PROGN (RAID "Invalid MODE for \DL.TRI.XFERDISK")))
                                                                                (* Fill in the virtual page field)
            (replace (TRIDENTIOCB TH.DATAPTRHI) of \IOCBPAGE with (\HILOC BUFFER))
            (replace (TRIDENTIOCB TH.DATAPTRLO) of \IOCBPAGE with (\LOLOC BUFFER))
                                                                    (* If we were given a label, we better put it on the IOCB page)
            (COND
               ((AND LABEL (NEQ MODE 'VRR)
                        (NEQ MODE 'VRW))
                     (\BLT (LOCF (fetch (TRIDENTIOCB TH.WRITEVERIFYLABEL) of \IOCBPAGE))
                          LABEL 10)))                              (* why the hell does the uCode need this???)
            (replace (TRIDENTIOCB TH.FILEPAGELO) of \IOCBPAGE with (\GETBASE (LOCF (fetch (TRIDENTIOCB TH.READLABEL)
                                                                                                of \IOCBPAGE))
                                                            5))
                                                            (* Finally, the moment has come..
                                                            Wake up the microcode)
                                                            (* The constant here removes uninteresting bits from the status)
            (replace (TRIDENTDCB MICRONEXT) of \IOPAGE with (\LOLOC \IOCBPAGE))
            (SETQ Status (LOGAND 56888 (\DL.TRI.INITIATE.IO)))

            (* * If we need to return a label, copy it out of the iocb page)

            (COND
               (LABEL (SELECTQ MODE
                           ((VRR VRW T)
                               (\BLT LABEL (LOCF (fetch (TRIDENTIOCB TH.READLABEL) of \IOCBPAGE))
                                    10))
                           NIL)))                                  (* only recal once)
            (replace (TRIDENTDCB RECALIBRATE) of \IOPAGE with NIL)  (* Trident controller indicates good completion in bit 2 from left
                                                                    of controller status)
            [COND
               ((NEQ (LOGAND 16384 Status)
                     0)
                (RETURN 'OK]                                        (* Convert status to be same as for shugart status)
            (SETQ RetryCount (SUB1 RetryCount))
            (COND
               ((EQ RetryCount 0)

            (* We have already recalibrated and tried ten more times and failed, so we're really stuck)

               (COND
                  [FLG (COND
                          (NORAIDFLG (RETURN Status))
                          (T (RAID "Hard disk error" Status]
                  (T [SETQ Status (LOGOR (COND
                                            ((NEQ (LOGAND Status 2048)
                                                  0)
                                             'VERIFYERROR))
                                         (COND
                                            ((NEQ (LOGAND Status 1024)
                                                  0)
                                             'READERROR))
                                         (COND
                                            ((NEQ (LOGAND Status 512)
                                                  0)
                                             'READERROR]
                     (RETURN Status)))                             (* Who knows? We might be lost.
                                                                   Recalibrate and try again)
               (replace (TRIDENTDCB RECALIBRATE) of \IOPAGE with T)
               (SETQ FLG T)
               (GO LP))
               (T (GO LP]
)

(DEFINEQ
```

(**DISKDLION.INIT**
```
  [LAMBDA NIL                                                      (* mpl " 2-Aug-85 15:39")
     (SETQ \DL.DOBPAGE (NCREATE 'VMEMPAGEP))
     (\LOCKPAGES \DL.DOBPAGE 1)
     (MAPC '(\MAKENUMBER \SETGLOBALVAL.UFN \RPLPTR.UFN \HTFIND \SLOWIPLUS2 \SLOWIDIFFERENCE \SLOWLLSH1 \SLOWLLSH8
                  \SLOWLRSH1 \SLOWLRSH8 \SLOWLOGOR2 \SLOWLOGAND2 \SLOWLOGXOR2 \SLOWIGREATERP \SLOWIQUOTIENT
                  \SLOWITIMES2 IMOD IREMAINDER)
           (FUNCTION \LOCKFN))
     (MAPC '(\DL.DISKINIT \DL.INIT.DOB.CHAIN \DL.ACTONVMEMFILE \DL.ACTONVMEMPAGE \DL.DISKSEEK \DL.XFERDISK
                  \DL.DISKOP \DL.RECALIBRATE \DL.SHUGART.XFERDISK \DL.OBTAINNEWDOB \DL.RELEASEDOB
                  \DLDISK.EXECUTE \DL.HANDLEMULTIRUN \DL.PUTDISKADDRESS \DL.TRI.INITIATE.IO \DL.TRI.XFERDISK)
           (FUNCTION \LOCKFN))
     (MAPC '(\SMALLNEGSPACE \IOCBPAGE \MDSTypeTable \HTCOLL \HTMAIN \VMBASEDP \DLDISKSHAPE.SECTORSPERHEAD
                  \DLDISKSHAPE.SECTORSPERCYLINDER \DLDISKSHAPE.HEADSPERDRIVE \DISKTYPE \CONTROLLERTYPE
                  \DL.DOBPAGE \DL.DOBCHAIN \DL.FAULTHANDLERDOB \DL.SPAREDOB)
           (FUNCTION \LOCKVAR])

)
```

```
(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQ? \DLDISKSHAPE.SECTORSPERCYLINDER NIL)

(RPAQ? \DLDISKSHAPE.HEADSPERDRIVE NIL)

(RPAQ? \DLDISKSHAPE.SECTORSPERHEAD NIL)

(RPAQ? \DL.FAULTHANDLERDOB NIL)

(RPAQ? \DL.DOBPAGE NIL)

(RPAQ? \DL.DOBCHAIN NIL)

(RPAQ? \DL.SPAREDOB NIL)

(RPAQ? \DISKTYPE NIL)

(RPAQ? \CONTROLLERTYPE NIL)

(\DISKDLION.INIT)
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ DISKDLDECLS
        ((FILES (SOURCE)
                DISKVMEMDECLS)
         (GLOBALVARS \DL.DOBCHAIN \DL.DOBPAGE \DLDISKSHAPE.SECTORSPERHEAD \DLDISKSHAPE.SECTORSPERCYLINDER
                \DLDISKSHAPE.HEADSPERDRIVE \DISKTYPE \CONTROLLERTYPE)
         (RECORDS DLDISK.STATUS TRIDENTIOCB)
         (MACROS \DLDISK.GETSTATUS)
         (FUNCTIONS \DL.DOB.FROM.PAGE.OFFSET \DL.PAGE.OFFSET.FROM.DOB)
         (CONSTANTS \DL.ABORT−NR/WF/OVR \DL.ABORT−NR/WF/OVR/CRC \DL.ABORT−NR/WF/OVR/CRC/VERIFY \DL.READFIELD
                \DL.SEEKIOCBSTART \DL.VERIFYFIELD \DL.WRITEFIELD \DL.XFERIOCBSTART \DL.MINUSSTEP \DL.PLUSSTEP)
         (CONSTANTS \DL.TRI.PASSWORD)
         (CONSTANTS \DL.DOBSIZE \DL.MAXDOBS)
         (RECORDS DLVMEMRUN IOCBPAGE)))

(FILESLOAD (SOURCE)
        DISKVMEMDECLS)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \DL.DOBCHAIN \DL.DOBPAGE \DLDISKSHAPE.SECTORSPERHEAD \DLDISKSHAPE.SECTORSPERCYLINDER
        \DLDISKSHAPE.HEADSPERDRIVE \DISKTYPE \CONTROLLERTYPE)
)

(DECLARE%: EVAL@COMPILE

[ACCESSFNS DLDISK.STATUS ((TRACK00 (NEQ (LOGAND DATUM 512)
                                        0))
                         (HEADSELECT (LOGXOR (LRSH DATUM 11)
                                        31))
                         (SA1000 (NEQ (LOGAND DATUM 32)
                                        0))
                         (DRIVENOTREADY (NEQ (LOGAND DATUM 16)
                                        0))
                         (WRITEFAULT (NEQ (LOGAND DATUM 8)
                                        0))
                         (OVERRUN (NEQ (LOGAND DATUM 4)
                                        0))
                         (CRCERR (NEQ (LOGAND DATUM 2)
                                        0))
                         (VERIFYERR (NEQ (LOGAND DATUM 1)
                                        0]

(BLOCKRECORD TRIDENTIOCB ((TH.CYL WORD)
                         (TH.HD BYTE)
                         (TH.SEC BYTE)
                         (TH.LABELPTRLO WORD)
                         (TH.LABELPTRHI WORD)
                         (TH.DATAPTRLO WORD)
                         (TH.DATAPTRHI WORD)
                         (TH.RETRIES WORD)
                         (TH.PAGECOUNT WORD)
                         (TH.CONTROLLERSTATUS WORD)
                         (TH.DISKSTATUS WORD)
                         (TH.uCYL WORD)
                         (TH.uHD BYTE)
                         (TH.uSEC BYTE)
                         (NIL WORD)
                         (TH.MICRONEXT WORD)
                         (TH.DCBOFFSET WORD)
                         (TH.FILEPAGELO WORD)
                         (TH.SEEK.XFERMASK WORD)
```

```
                              (TH.SEEK.KDRIVE WORD)
                              (TH.SEEK.KCYL WORD)
                              (TH.SEEK.KHEAD WORD)
                              (TH.HEADER.OP WORD)
                              (TH.HEADER.COMMAND WORD)
                              (TH.HEADER.CONTROL WORD)
                              (TH.HEADER.ERRORMASK WORD)
                              (TH.LABEL.OP WORD)
                              (TH.LABEL.COMMAND WORD)
                              (TH.LABEL.CONTROL WORD)
                              (TH.LABEL.ERRORMASK WORD)
                              (TH.DATA.OP WORD)
                              (TH.DATA.COMMAND WORD)
                              (TH.DATA.CONTROL WORD)
                              (TH.DATA.ERRORMASK WORD)
                              (TH.WRITEVERIFYLABEL 10 WORD)
                              (NIL 5 WORD)
                              (TH.HDSFROMINITIAL WORD)
                              (TH.READLABEL 10 WORD)
                              (NIL 8 WORD)
                              (TH.DSB0.PILOTNEXT WORD)
                              (TH.DSB0.MICRONEXT WORD)
                              (TH.DSB0.PILOTTAIL WORD)
                              (TH.DSB0.SELECTDATA WORD)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \DLDISK.GETSTATUS MACRO (NIL (\DEVICE.INPUT 3)))
)


(DEFMACRO \DL.DOB.FROM.PAGE.OFFSET (OFFSET)
    `(\ADDBASE \DL.DOBPAGE ,OFFSET))


(DEFMACRO \DL.PAGE.OFFSET.FROM.DOB (DOB)
    `(IDIFFERENCE (\LOLOC ,DOB)
           (\LOLOC \DL.DOBPAGE)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \DL.ABORT-NR/WF/OVR 28)

(RPAQQ \DL.ABORT-NR/WF/OVR/CRC 30)

(RPAQQ \DL.ABORT-NR/WF/OVR/CRC/VERIFY 31)

(RPAQQ \DL.READFIELD 1072)

(RPAQQ \DL.SEEKIOCBSTART 338)

(RPAQQ \DL.VERIFYFIELD 1074)

(RPAQQ \DL.WRITEFIELD 1083)

(RPAQQ \DL.XFERIOCBSTART 355)

(RPAQQ \DL.MINUSSTEP 1120)

(RPAQQ \DL.PLUSSTEP 1056)

(CONSTANTS \DL.ABORT-NR/WF/OVR \DL.ABORT-NR/WF/OVR/CRC \DL.ABORT-NR/WF/OVR/CRC/VERIFY \DL.READFIELD
      \DL.SEEKIOCBSTART \DL.VERIFYFIELD \DL.WRITEFIELD \DL.XFERIOCBSTART \DL.MINUSSTEP \DL.PLUSSTEP)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \DL.TRI.PASSWORD 279)

(CONSTANTS \DL.TRI.PASSWORD)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \DL.DOBSIZE 16)

(RPAQQ \DL.MAXDOBS 14)

(CONSTANTS \DL.DOBSIZE \DL.MAXDOBS)
)

(DECLARE%: EVAL@COMPILE

[BLOCKRECORD DLVMEMRUN ((DLFIRSTFILEPAGE WORD)
                        (DLVMCYL WORD)
                        (DLVMHEAD BYTE)
```

```
                              (DLVMSECTOR BYTE))
              (ACCESSFNS DLVMEMRUN ((DLNEXTRUN (\ADDBASE DATUM 3]

(BLOCKRECORD IOCBPAGE ((NIL 3 WORD)
                       (LASTIOCBSTATUS WORD)
                       (NEXTIOCB WORD)
                       (SEEKIOCBLOC WORD)
                       (XFERIOCBLOC WORD)
                       (VRRIOCBLOC WORD)
                       (VVRIOCBLOC WORD)
                       (HCYLINDER WORD)
                       (HHEAD BYTE)
                       (HSECTOR BYTE)
                       (LID 5 WORD)
                       (LPAGELO WORD)
                       (LPAGEHI BITS 7)
                       (NIL BITS 6)
                       (LFLAGS BITS 3)
                       (LTYPE WORD)
                       (LBOOTLINKCHAIN1 WORD)
                       (LBOOTLINKCHAIN2 WORD)
                       (NIL 43 WORD)
                       (PRUNLENGTH WORD)
                       (NIL 6 WORD)
                       (PLABELCMD WORD)
                       (PLABELLEN WORD)
                       (NIL WORD)
                       (PLABELABORT WORD)
                       (PDATACMD WORD)
                       (PDATALEN WORD)
                       (PVPAGE WORD)
                       (PDATAABORT WORD)
                       (PTERMCOND1HEAD BITS 5)
                       (PTERMCOND1MAGIC BITS 11)
                       (PTERMCOND2HEAD BITS 5)
                       (PTERMCOND2MAGIC BITS 11)
                       (SCYLINDERDISPLACEMENT WORD)
                       (NIL 5 WORD)
                       (SSEEKCMD1 WORD)
                       (NIL WORD)
                       (SSEEKCMD2 WORD)
                       (NIL 9 WORD)
                       (NIL 20 WORD)                    (* Used to be 21, but that bumps into vmem file info.
                                                        RS232 bug?)

                       (RS232CGETIOCB 8 WORD)
                       (DLVMEMFILEINFO 113 WORD)         (* bunch of VMEMFILERUN entries describing vmem disk
                                                        addresses)

                       (FLOPPYIOCBSTART 16 WORD)))
)

(RPAQQ TRIDISKDLDECLS
       ((RECORDS TRIDENTDCB TRIDENTIOCB)
        (CONSTANTS \TH.DR.COMMAND \TH.DR.CONTROL \TH.DR.ERRORMASK \TH.DR.OP \TH.DW.COMMAND \TH.DW.CONTROL
                \TH.DW.ERRORMASK \TH.DW.OP \TH.LN.COMMAND \TH.LN.CONTROL \TH.LN.ERRORMASK \TH.LN.OP
                \TH.LR.COMMAND \TH.LR.CONTROL \TH.LR.ERRORMASK \TH.LR.OP \TH.LV.COMMAND \TH.LV.CONTROL
                \TH.LV.ERRORMASK \TH.LV.OP \TH.LW.COMMAND \TH.LW.CONTROL \TH.LW.ERRORMASK \TH.LW.OP
                \TH.WW.CONTROL \DL.TRI.STARTCMD \TH.SEEK.HDCMD \TH.SEEK.CYLCMD)
        (CONSTANTS \TH.HV.COMMAND \TH.HV.CONTROL \TH.HV.ERRORMASK \TH.HV.OP \DL.TRI.PASSWORD)
        (CONSTANTS \TH.DOCTL.RESETDEVICECHECK \TH.DOCTL.SETHD5 \TH.NULLCTL.RESETDEVICECHECK \TH.NULLCTL.SETHD5))
)

(DECLARE%: EVAL@COMPILE

(BLOCKRECORD TRIDENTDCB ((PILOTNEXT WORD)
                         (MICRONEXT WORD)
                         (PILOTTAIL WORD)
                         (RECALIBRATE FLAG)
                         (SELECTMASK BITS 7)
                         (DISKTYPE BITS 3)
                         (DCBOFFSET BITS 4)))

(BLOCKRECORD TRIDENTIOCB ((TH.CYL WORD)
                          (TH.HD BYTE)
                          (TH.SEC BYTE)
                          (TH.LABELPTRLO WORD)
                          (TH.LABELPTRHI WORD)
                          (TH.DATAPTRLO WORD)
                          (TH.DATAPTRHI WORD)
                          (TH.RETRIES WORD)
                          (TH.PAGECOUNT WORD)
                          (TH.CONTROLLERSTATUS WORD)
                          (TH.DISKSTATUS WORD)
                          (TH.uCYL WORD)
                          (TH.uHD BYTE)
                          (TH.uSEC BYTE)
                          (NIL WORD)
                          (TH.MICRONEXT WORD)
```

```
                              (TH.DCBOFFSET WORD)
                              (TH.FILEPAGELO WORD)
                              (TH.SEEK.XFERMASK WORD)
                              (TH.SEEK.KDRIVE WORD)
                              (TH.SEEK.KCYL WORD)
                              (TH.SEEK.KHEAD WORD)
                              (TH.HEADER.OP WORD)
                              (TH.HEADER.COMMAND WORD)
                              (TH.HEADER.CONTROL WORD)
                              (TH.HEADER.ERRORMASK WORD)
                              (TH.LABEL.OP WORD)
                              (TH.LABEL.COMMAND WORD)
                              (TH.LABEL.CONTROL WORD)
                              (TH.LABEL.ERRORMASK WORD)
                              (TH.DATA.OP WORD)
                              (TH.DATA.COMMAND WORD)
                              (TH.DATA.CONTROL WORD)
                              (TH.DATA.ERRORMASK WORD)
                              (TH.WRITEVERIFYLABEL 10 WORD)
                              (NIL 5 WORD)
                              (TH.HDSFROMINITIAL WORD)
                              (TH.READLABEL 10 WORD)
                              (NIL 8 WORD)
                              (TH.DSB0.PILOTNEXT WORD)
                              (TH.DSB0.MICRONEXT WORD)
                              (TH.DSB0.PILOTTAIL WORD)
                              (TH.DSB0.SELECTDATA WORD)))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \TH.DR.COMMAND 11332)

(RPAQQ \TH.DR.CONTROL 61582)

(RPAQQ \TH.DR.ERRORMASK 1536)

(RPAQQ \TH.DR.OP 14)

(RPAQQ \TH.DW.COMMAND 11396)

(RPAQQ \TH.DW.CONTROL 61583)

(RPAQQ \TH.DW.ERRORMASK 512)

(RPAQQ \TH.DW.OP 13)

(RPAQQ \TH.LN.COMMAND 15364)

(RPAQQ \TH.LN.CONTROL 61588)

(RPAQQ \TH.LN.ERRORMASK 512)

(RPAQQ \TH.LN.OP 0)

(RPAQQ \TH.LR.COMMAND 11332)

(RPAQQ \TH.LR.CONTROL 61580)

(RPAQQ \TH.LR.ERRORMASK 1536)

(RPAQQ \TH.LR.OP 10)

(RPAQQ \TH.LV.COMMAND 11332)

(RPAQQ \TH.LV.CONTROL 61573)

(RPAQQ \TH.LV.ERRORMASK 3584)

(RPAQQ \TH.LV.OP 11)

(RPAQQ \TH.LW.COMMAND 11396)

(RPAQQ \TH.LW.CONTROL 61581)

(RPAQQ \TH.LW.ERRORMASK 512)

(RPAQQ \TH.LW.OP 9)

(RPAQQ \TH.WW.CONTROL 61455)

(RPAQQ \DL.TRI.STARTCMD 3072)

(RPAQQ \TH.SEEK.HDCMD 3072)

(RPAQQ \TH.SEEK.CYLCMD 3072)
```

```
(CONSTANTS \TH.DR.COMMAND \TH.DR.CONTROL \TH.DR.ERRORMASK \TH.DR.OP \TH.DW.COMMAND \TH.DW.CONTROL
        \TH.DW.ERRORMASK \TH.DW.OP \TH.LN.COMMAND \TH.LN.CONTROL \TH.LN.ERRORMASK \TH.LN.OP \TH.LR.COMMAND
        \TH.LR.CONTROL \TH.LR.ERRORMASK \TH.LR.OP \TH.LV.COMMAND \TH.LV.CONTROL \TH.LV.ERRORMASK \TH.LV.OP
        \TH.LW.COMMAND \TH.LW.CONTROL \TH.LW.ERRORMASK \TH.LW.OP \TH.WW.CONTROL \DL.TRI.STARTCMD \TH.SEEK.HDCMD
        \TH.SEEK.CYLCMD)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \TH.HV.COMMAND 11332)

(RPAQQ \TH.HV.CONTROL 63619)

(RPAQQ \TH.HV.ERRORMASK 3584)

(RPAQQ \TH.HV.OP 279)

(RPAQQ \DL.TRI.PASSWORD 279)

(CONSTANTS \TH.HV.COMMAND \TH.HV.CONTROL \TH.HV.ERRORMASK \TH.HV.OP \DL.TRI.PASSWORD)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \TH.DOCTL.RESETDEVICECHECK 9224)

(RPAQQ \TH.DOCTL.SETHD5 17413)

(RPAQQ \TH.NULLCTL.RESETDEVICECHECK 1032)

(RPAQQ \TH.NULLCTL.SETHD5 1029)

(CONSTANTS \TH.DOCTL.RESETDEVICECHECK \TH.DOCTL.SETHD5 \TH.NULLCTL.RESETDEVICECHECK \TH.NULLCTL.SETHD5)
)
)

(PUTPROPS DISKDLION COPYRIGHT ("Venue & Xerox Corporation" 1984 1985 1986 1987 1990))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX

MACRO INDEX