

File created: 11-May-2023 21:48:37 {DSK}<cygdrive>c>Users>Larry>home>il>MEDLEY>SOURCES>MOD44IO.;2

edit by: lmm

changes to: (VARS MOD44IOCOMS)

previous date: 16-Mar-2021 19:55:51 {DSK}<cygdrive>c>Users>Larry>home>il>MEDLEY>SOURCES>MOD44IO.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ MOD44IOCOMS

(

;;; Dorado disk driver

(COMS ;; Device dependent code for the Model44 disk

(FNS \M44AddDiskPages \M44CloseFile \M44CompleteFH \M44CREATEFILE \M44DeleteFile \M44EVENTFN
\M44ExtendFilePageMap \M44FillInMap \M44GetFileHandle \M44GetFileInfo \M44GETDATEPROP
\M44GetFileName \M44GetPageLoc \M44KillFilePageMap \M44MAKEDIRENTRY \M44OpenFile
\M44OPENFILEFROMFP \M44ReadDiskPage \M44ReadLeaderPage \M44ReadPages \M44SetAccessTimes
\M44SetEndOfFile \M44SetFileInfo \M44SETFILETYPE \M44TruncateFile \M44WriteDiskPage
\M44WriteLeaderPage \M44WritePages \M44WritePages1))

(COMS ;; Disk allocation

(FNS \ADDDISKPAGES \M44DELETEPAGES \ASSIGNDISKPAGE \COUNTDISKFREEPAGES \M44MARKPAGEFREE
\M44FLUSHDISKDESCRIPTOR \MAKELEADERDAS DISKFREEPAGES \M44FREEPAGECOUNT))

(COMS (INITVARS (\M44MULTFLG T))

(DECLARE%: DONTCOPY (MACROS UCASECHAR UPDATEVALIDATION)
(RECORDS M44DEVICE)
(GLOBALVARS \M44MULTFLG \DISKNAMECASEARRAY)
(MACROS .LISP.TO.BFS. .BFS.TO.LISP. .DISKCASEARRAY.)
(CONSTANTS (PageMapIncrement 64)
(\MAX.ALTO.NAME.LENGTH 39))

(COMS ;; File properties

(RECORDS M44FILEPROP)
(CONSTANTS * FPROPTYPES)
(CONSTANTS * FPTYPES))
(GLOBALRESOURCES \M44PAGEBUFFER))
(INITRESOURCES \M44PAGEBUFFER))

(COMS ;; Directory enumeration

(FNS \M44GENERATEFILES \M44SORTFILES \M44GENERATENEXT \M44NEXTFILEFN \M44SORTEDNEXTFILEFN
\M44FILEINFOFN))

(COMS ;; Directory lookup routines

(FNS \M44PARSEFILENAME \FINDDIRHOLE \M44PACKFILENAME \M44READVERSION \OPENDISKDESCRIPTOR
\M44READDIRFID \M44READDIRNAME \M44SEARCHDIR \M44UNPACKFILENAME)
(VARS \FILENAMECHARSLST)
(GLOBALVARS \FILENAMECHARSLST)
(DECLARE%: DONTCOPY (RECORDS UNAME FILESPEC M44GENFILESTATE M44DIRSEARCHSTATE)
(MACROS BETWEEN)))

(COMS (FNS \CREATE.FID.FOR.DD \OPENDISK \OPENDISKDEVICE \OPENDIR \M44CHECKPASSWORD \M44HOSTNAMEP)
(DECLARE%: DONTCOPY (CONSTANTS \OFFSET.BCPLUSUSERNAME \OFFSET.BCPLPASSWORD \NWORDS.BCPLPASSWORD)))

[COMS ;; SYSOUT etc.

(FNS \COPYSYS1)

;; For MAIKO. \COPYSYS use UNIX-PAGEPERBLOCK.

(FNS \MAIKO.CHECKFREESPACE)
(INITVARS (\LDEDESTOVERWRITE NIL))
(DECLARE%: DONTCOPY (CONSTANTS (LISPPAGE-PER-UNIXBLOCK 2)

(COMS ;; Stats code. On MOD44IO because it writes on the disk and uses records not exported from MOD44IO. (For this and other
;; reasons, GATHERSTATS only works on Dorados.)

(FNS GATHERSTATS)
(VARS (\STATSON NIL)))
(DECLARE%: EVAL@COMPILE DONTCOPY (LOCALVARS . T)
(FILES (LOADCOMP
LLBFS))))

;;; Dorado disk driver

;; Device dependent code for the Model44 disk

(DEFINEQ

(\M44AddDiskPages

[LAMBDA (STREAM NEWLASTPAGE NEWLASTBYTE)

; Edited 21-Jan-91 23:35 by jds

;; Add pages to an existing Model44 file. NEWLASTPAGE is the page number of the last page in the extended file. Return the disk address of the
;; new last page.

```
(M44FillInMap STREAM (fetch (M44STREAM LastPage) of STREAM)) ; Fill in map to end of file. Code below assumes at least one
; valid map entry
(\ADDDISKPAGES STREAM (ADD1 (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM))
(IDIFFERENCE NEWLASTPAGE (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM))
(fetch (ARRAYP BASE) of (\M44ExtendFilePageMap STREAM NEWLASTPAGE))
NEWLASTBYTE)
(replace (M44STREAM LASTMAPPEDPAGE) of STREAM with NEWLASTPAGE)
(replace (M44STREAM LastPage) of STREAM with NEWLASTPAGE)
(replace (M44STREAM LastOffset) of STREAM with NEWLASTBYTE) ; record new eof in filehandle only
NEWLASTPAGE])
```

(M44CloseFile

```
[LAMBDA (STREAM) ; (* hdj "25-Sep-86 11:03")
(\CLEARMAP STREAM)
[COND
((NEQ (fetch ACCESS of STREAM)
'INPUT) ; Update EOF in leader page
(\M44TruncateFile STREAM (fetch EPAGE of STREAM)
(fetch EOFFSET of STREAM)
T)
(\M44FLUSHDISKDESCRIPTOR (fetch DEVICE of STREAM]
STREAM])
```

(M44CompleteFH

```
[LAMBDA (STREAM) ; Edited 21-Jan-91 23:41 by jds
;; Completes the fields of a file handle that describes an existing file by reading in its leader page which it leaves for its caller
(PROG ((NUMCHARS (CONS))
(LEADERPAGE (\M44ReadLeaderPage STREAM))
(DSK (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM)))
LASTPAGE# NBYTES)
;; Get the page number and the number of bytes on the last page of the file specified by fHandle. If the last page number hint is wrong in the
;; leader page, then find the real last page and change the hint.
(COND
((AND (NEQ (SETQ LASTPAGE# (.BFS.TO.LISP. (fetch (\M44LeaderPage LastPageNumber) of LEADERPAGE)))
-1)
(EQ [PROG ((DAs (ARRAY 3 'WORD \FILLINDA 0))
(BFSPG# (.LISP.TO.BFS. LASTPAGE#)))
(SETA DAs 1 (fetch (\M44LeaderPage LastPageAddress) of LEADERPAGE))
(SETA DAs 2 \EOFDA)
(RETURN (AND (EQ (\ACTONDISKPAGES DSK NIL (fetch (ARRAYP BASE) of DAs)
LASTPAGE# STREAM BFSPG# BFSPG# \DC.READD NUMCHARS NIL T)
BFSPG#)
(SETQ NBYTES (CAR NUMCHARS)
(fetch (\M44LeaderPage LastPageByteCount) of LEADERPAGE)))
(replace (M44STREAM LastPage) of STREAM with LASTPAGE#) ; Update STREAM eof
(replace (M44STREAM LastOffset) of STREAM with NBYTES))
(T ; Hint was wrong so scan the file for last page
(for PN from PageMapIncrement by PageMapIncrement do (SETQ LASTPAGE# (\M44FillInMap STREAM PN))
; Wait until attempt to find page fails
repeatwhile (EQ PN LASTPAGE#))
(SETQ NBYTES (\ACTONDISKPAGES DSK NIL (fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP)
of STREAM))
-1 STREAM (.LISP.TO.BFS. LASTPAGE#)
(.LISP.TO.BFS. LASTPAGE#)
\DC.READD NUMCHARS)) ; Read last page to find out how many bytes are on it
(\M44SetEndOfFile STREAM LASTPAGE# (CAR NUMCHARS)
T)))
(UPDATEVALIDATION STREAM LEADERPAGE) ; Validation is low order bits of creation and write dates
[COND
((EQ (fetch (M44STREAM LastOffset) of STREAM)
BYTESPERPAGE)
;; Shouldn't happen, because alto files should never have a full last page. However, cope if it happens...
(replace EPAGE of STREAM with (ADD1 (fetch (M44STREAM LastPage) of STREAM)))
(replace EOFFSET of STREAM with 0))
(T (replace EPAGE of STREAM with (fetch (M44STREAM LastPage) of STREAM))
(replace EOFFSET of STREAM with (fetch (M44STREAM LastOffset) of STREAM))
(RETURN STREAM])
```

(M44CREATEFILE

```
[LAMBDA (FDEV UNAME LENGTH CRDATE TYPE DIRECTORYP) ; Edited 21-Jan-91 23:41 by jds
;; Create a file on the Model44 disk.
(PROG ((DSK (fetch (M44DEVICE DSKOBJ) of FDEV))
(PNAME (\M44PACKFILENAME UNAME))
(LEADERPAGE (create \M44LeaderPage))
(NC 0)
STREAM FP MAP FPBASE DAT PSTART)
(OR PNAME (RETURN)) ; Cant create as name wasnt complete
(SETQ STREAM (create M44STREAM))
(replace FULLFILENAME of STREAM with PNAME)
```

```
(replace DEVICE of STREAM with FDEV)
(replace (M44STREAM FID) of STREAM with (SETQ FP (create FID)))
(replace (M44STREAM FILEPAGEMAP) of STREAM with (SETQ MAP (ARRAY (COND
    ((FIXP LENGTH)
     (IPLUS 4 (FOLDHI LENGTH
              BYTESPERPAGE)))
    (T PageMapIncrement))
    'WORD \FILLINDA 0)))

(replace (M44STREAM LASTMAPPEDPAGE) of STREAM with 0)
(replace MULTIBUFFERHINT of STREAM with \M44MULTFLG)
(replace (M44STREAM LEADERPAGE) of STREAM with LEADERPAGE)
(SETQ FPBASE (fetch (ARRAYP BASE) of FP))
(replace (FP FPSERIAL#) of FPBASE with (add (fetch (DSKOBJ DISKLASTSERIAL#) of DSK
    1)))

(COND
  (DIRECTORYP (add (fetch (FP FPSERIALHI) of FPBASE)
    \FP.DIRECTORYP)))
(replace (FP FPVERSION) of FPBASE with 1)
(SETA MAP 0 \EOFDA)
(SETA MAP 3 \EOFDA)
; We are about to create pages 0 and 1, everything else is
; nonexistent
; Done by the NCREATE -- (\ZEROPAGE (fetch (POINTER
; PAGE#) of LEADERPAGE))

(\BLT (LOCF (fetch (\M44LeaderPage TimeWrite) of LEADERPAGE))
  (SETQ DAT (\DAYTIME0 (create FIXP)))
  WORDSPERCELL) ; Set creation and write dates
(\BLT (LOCF (fetch (\M44LeaderPage TimeCreate) of LEADERPAGE))
  (OR CRDATE DAT)
  WORDSPERCELL) ; See \M44MAKEDIRENTRY for the name logic.
(for C in (fetch (UNAME ORIGCHARS) of UNAME) bind (NAMEBASE _ (LOCF (fetch (\M44LeaderPage NameCharCount)
  of LEADERPAGE)))
  (V _ (fetch (UNAME VERSION) of UNAME))

  do (\PUTBASEBYTE NAMEBASE (add NC 1)
    C)
  finally [COND
    ((NEQ V 1)
     (\PUTBASEBYTE NAMEBASE (add NC 1)
      (CHARCODE !))
     (for C in (CHCON V) do (\PUTBASEBYTE NAMEBASE (add NC 1)
       C]
     (\PUTBASEBYTE NAMEBASE (add NC 1)
      (CHARCODE %.)
      ; Last character of all alto names is dot
    (replace (\M44LeaderPage NameCharCount) of LEADERPAGE with NC))
(replace (\M44LeaderPage PropertyBegin) of LEADERPAGE with (INDEXF (FETCH (\M44LeaderPage LeaderProps)
  of LEADERPAGE)))
[replace (\M44LeaderPage PropertyLength) of LEADERPAGE with (CONSTANT (- (INDEXF (FETCH (\M44LeaderPage
  Spares)
  of LEADERPAGE))
  (INDEXF (FETCH (\M44LeaderPage
  LeaderProps)
  of LEADERPAGE]
; The start and length of the property section are theoretically
; variable, but at least some "official" Alto software, such as
; Scavage, believes that file names must be no more than 39
; chars.

(\M44SETFILETYPE STREAM TYPE)
(\WRITEDISKPAGES DSK (LIST LEADERPAGE NIL)
  (fetch (ARRAYP BASE) of MAP)
  -1 STREAM 0 1 NIL NIL 0 0)
; The end of file will be zero and the validation not set as befits a
; new file.

(replace (FP FPLEADERVDA) of FPBASE with (\WORDELT MAP 1))
; Now that the file is safely created, make entry in directory
(replace (M44STREAM DIRINFO) of STREAM with (\M44MAKEDIRENTRY (fetch (M44STREAM FID) of STREAM)
  UNAME NC FDEV))
(RETURN STREAM)]
```

(M44DeleteFile

```
[LAMBDA (FILENAME DEV)
; Edited 21-Jan-91 23:35 by jds
; Delete a Model44 file.

(PROG ((STREAM (M44GetFileHandle FILENAME 'OLDEST DEV T))
(COND
  ((OR (NOT STREAM)
    (FDEVOP 'OPENP DEV (fetch FULLFILENAME of STREAM)
    NIL DEV))
    ; Can't delete an open file
  (RETURN)))
(\M44DELETEPAGES STREAM -1)
(PROG ((DIROFD (fetch (M44DEVICE SYSDIROFD) of DEV))
  (\SETFILEPTR DIROFD (fetch (M44STREAM DIRINFO) of STREAM))
  (\ABOUT DIROFD (LOGAND 3 (\PEEKBIN DIROFD)))
  (FLUSHMAP DIROFD))
  ; Delete directory entry
(\M44KillFilePageMap STREAM)
(replace (M44STREAM FID) of STREAM with NIL)
(RETURN (fetch FULLFILENAME of STREAM))
```

(M44EVENTFN

```
[LAMBDA (FDEV EVENT) ; Edited 21-Jan-91 23:31 by jds
  (DECLARE (GLOBALVARS \DEVICENAMETODEVICE \MACHINETYPE))
  (SELECTQ EVENT
    ((AFTERLOGOUT AFTERSYSOUT AFTERMAKESYS AFTERSAVEVM) ;; reinitialize DSK device and revalidate its open streams
      [PROG ((DSKOBJ (fetch (M44DEVICE DSKOBJ) of FDEV)
        DD)
          (COND
            ((SETQ DD (fetch (DSKOBJ DISKDESCRIPTOROFD) of DSKOBJ))
              ; Flush out of date disk descriptor
              (FORGETPAGES DD)
              (FDEVOP 'UNREGISTERFILE FDEV FDEV DD) ; Stream no longer in use. Don't go thru \M44CloseFile because
              ; it will try to Truncate, etc.
              (replace (DSKOBJ DDVALID) of DSKOBJ with NIL)
              (replace (DSKOBJ DISKDESCRIPTOROFD) of DSKOBJ with NIL)))
            (FORGETPAGES (fetch (DSKOBJ SYSDIROFD) of DSKOBJ))
            (FDEVOP 'UNREGISTERFILE FDEV FDEV (fetch (DSKOBJ SYSDIROFD) of DSKOBJ))
            (COND
              [(AND (EQ \MACHINETYPE \DORADO)
                (LET [(PARTZEROP (EQ (fetch (M44DEVICE DSKPARTITION) of FDEV)
                  0))
                  (CURPARTP (EQ (fetch (FDEV DEVICENAME) of FDEV)
                    (PACK* 'DSK (DISKPARTITION)]
                  (COND
                    (PARTZEROP ; This is interlock with \M44EXTENDVMMEMFILE which doesn't
                    ; want to mess up the DiskDescriptor
                    (SETQ \M44.READY T)))
                  (COND
                    ((OR (AND PARTZEROP CURPARTP)
                      (\DEVICE-OPEN-STREAMS FDEV))
                    (COND
                      ((EQ PARTZEROP CURPARTP) ; No partition change to worry about, just reopen dir
                      (\OPENDIR FDEV))
                      (PARTZEROP
                        ;; This was the default partition, no longer is, so reopen it as if from scratch. Also,
                        ;; remove the mapping of DSK to this device
                        (\REMOVEDEVICE.NAMES FDEV 'DSK)
                        (\OPENDISK (SUBATOM (fetch (FDEV DEVICENAME) of FDEV)
                          4)
                          FDEV))
                        (T ; This was a non-default partition, now the default. Reopen it
                          ; with \MAINDISK as its DSKOBJ
                          (\OPENDISKDEVICE NIL NIL FDEV]
                    (T ; Device no longer exists if machine is now Dandelion; and if there were no open files, no need to try reopening the
                    ;; dir
                    (replace (DSKOBJ SYSDIROFD) of DSKOBJ with NIL)
                    ;; Have to explicitly clear these fields, because when we drop the DSKOBJ on the floor, GC does not know about its
                    ;; POINTER fields
                    (replace REOPENFILE of FDEV with (FUNCTION NIL))
                    ; In case there are files open over sysout as we come back on
                    ; Dandelion
                    (\REMOVEDEVICE FDEV]
          (\PAGED.REVALIDATEFILELST FDEV))
  (BEFORELOGOUT (\FLUSH.OPEN.STREAMS FDEV)
    (\M44FLUSHDISKDESCRIPTOR FDEV))
  NIL])
```

(\M44ExtendFilePageMap

```
[LAMBDA (STREAM TOPAGE#) ; Edited 21-Jan-91 23:35 by jds
  ;; If the file's page map is not big enough to map the given page, then create a new one that is big enough and copy the old OLDMAP information
  ;; into the new map. If the file has no map, then create one big enough to map the given page. Return the new map. --- Map entry 0 corresponds
  ;; to bfs page -1, entry 1 corresponds to the leader page, and entry 2 corresponds to Lisp page 0
  (PROG ((OLDMAP (fetch (M44STREAM FILEPAGEMAP) of STREAM)
    OLDSize NEWMAP)
    (RETURN (COND
      [(AND OLDMAP (ILESSP (IPLUS TOPAGE# 3)
        (SETQ OLDSize (fetch (ARRAYP LENGTH) of OLDMAP]
        OLDMap)
        (T (SETQ NEWMAP (ARRAY (CEIL (IPLUS TOPAGE# 4)
          PageMapIncrement)
          'SMALLPOSP \FILLINDA 0))
        (COND
          (OLDMap ; Copy old map into new
            (\BLT (fetch (ARRAYP BASE) of NEWMAP)
              (fetch (ARRAYP BASE) of OLDMap)
              OLDSize))
          (T ; Initialize with leader page hint
            (SETA NEWMAP 0 \EOFDA)
            (SETA NEWMAP 1 (fetch (FP FLEADERVDA) of (fetch (ARRAYP BASE)
              of (fetch (M44STREAM FID) of STREAM]
            (replace (M44STREAM FILEPAGEMAP) of STREAM with NEWMAP)
            NEWMAP]))
```

(M44FillInMap

[LAMBDA (STREAM UPTOPAGE)

; Edited 21-Jan-91 23:42 by jds

:: Assures that the disk address map for STREAM is filled in up thru page# UPTOPAGE. Reads file as needed

```
(PROG ( (MAP (\M44ExtendFilePageMap STREAM UPTOPAGE))
  (DSK (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM)))
  (LASTKNOWNPAGE (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM))
  NPAGES LASTPAGEREAD LASTATTEMPTED DAs DA) ; Extend MAP
  (SETQ DAs (fetch (ARRAYP BASE) of MAP))
  [while (ILESSP LASTKNOWNPAGE UPTOPAGE) do (COND
    [(NEQ (SETQ DA (\GETBASE DAs (IPLUS LASTKNOWNPAGE 1 2)))
      \FILLINDA)
      ; There already is an entry for the next page, so no need to read
      ; it
      (COND
        ((EQ DA \EOFDA)
          (RETURN))
        (T (add LASTKNOWNPAGE 1]
          (T [SETQ NPAGES (IMIN \MAXDISKDA (ADD1 (IDIFFERENCE
            UPTOPAGE
            LASTKNOWNPAGE
            ]
            ; We know where LASTKNOWNPAGE lives, so read it to find out where the next page after that is.
            ; Can do this for many pages at once to make it reasonable
            (SETQ LASTPAGEREAD
              (\ACTONDISKPAGES DSK NIL DAs -1 STREAM (.LISP.TO.BFS.
                LASTKNOWNPAGE)
              [SETQ LASTATTEMPTED
                (.LISP.TO.BFS. (SUB1 (IPLUS LASTKNOWNPAGE
                  NPAGES]
                \DC.READD))
              (SETQ LASTKNOWNPAGE (.BFS.TO.LISP. LASTPAGEREAD))
              (COND
                ((ILESSP LASTPAGEREAD LASTATTEMPTED)
                  ; Hit end of file
                  (RETURN]
                (replace (M44STREAM LASTMAPPEDPAGE) of STREAM with LASTKNOWNPAGE)
                (RETURN LASTKNOWNPAGE])
```

(M44GetFileHandle

[LAMBDA (NAME RECOG FDEV FAST CREATEFLG)

; Edited 21-Jan-91 23:48 by jds

:: Creates a STREAM for disk file NAME. If file does not exist, but CREATEFLG is true, returns the UNAME of the file so that it may be created. If FAST is true, does not fill in any fields of STREAM that would require reading the file, e.g., the length and full map

```
(LET ((DIRSTREAM (fetch (M44DEVICE SYSDIROFD) of FDEV))
  FS DP STREAM)
  (COND
    ((NULL DIRSTREAM) ; Non-existent device
      NIL)
    ((NULL (SETQ FS (\M44PARSEFILENAME NAME RECOG FDEV CREATEFLG)))
      ; File not found
      NIL)
    ((SETQ DP (fetch (FILESPEC FSDIRPTR) of FS))
      ; File was found--here's the directory pointer
      (SETQ STREAM (create M44STREAM))
      (replace DEVICE of STREAM with FDEV)
      (replace (M44STREAM FID) of STREAM with (\M44READDIRFID DIRSTREAM DP))
      (replace (M44STREAM DIRINFO) of STREAM with DP)
      (replace FULLFILENAME of STREAM with (\M44PACKFILENAME (fetch (FILESPEC UNAME) of FS)
        DP DIRSTREAM))
      (replace MULTIBUFFERHINT of STREAM with \M44MULTIFLG)
      (OR FAST (\M44CompleteFH STREAM))
      STREAM)
    ((NULL (fetch (FILESPEC UNAME) of FS))
      ; Name was malformed--can't create it even if we want to
      (LISPERROR "BAD FILE NAME" NAME))
    (CREATEFLG (fetch (FILESPEC UNAME) of FS])
```

(M44GetFileInfo

[LAMBDA (STREAM ATTRIBUTE DEV)

; Edited 21-Jan-91 23:44 by jds

:: Get the value of the ATTRIBUTE for a model44 file. If STREAM is a filename, then the file is not open.

```
(COND
  ((OR (type? STREAM STREAM)
    (SETQ STREAM (\M44GetFileHandle STREAM 'OLD DEV T)))
    (SELECTQ ATTRIBUTE
      ((LENGTH SIZE)
        (COND
          ((NULL (fetch VALIDATION of STREAM))
            ; Need to read leader page etc to get length
            (\M44CompleteFH STREAM))
          (SELECTQ ATTRIBUTE
            (LENGTH (create BYTEPTR
              PAGE _ (fetch EPAGE of STREAM)
```

```

      OFFSET _ (fetch EOFFSET of STREAM)))
  (IPLUS (fetch EPAGE of STREAM)
    (FOLDHI (fetch EOFFSET of STREAM
      BYTESPERPAGE)))
  (TYPE [PROG ((BUF (\M44ReadLeaderPage STREAM))
    (RETURN (COND
      ((IGREATERP (fetch (\M44LeaderPage PropertyLength) of BUF)
        0)
        (SETQ BUF (\ADDBASE BUF (fetch (\M44LeaderPage PropertyBegin) of BUF)))
        (do (SELECTC (fetch (M44FILEPROP FPROPTYPE) of BUF)
          (0
            (RETURN))
            ; End of properties
            (\FPROP.TYPE [RETURN (SELECTC (fetch (M44FILEPROP FPROPWORD0)
              of BUF)
                (\FPTYPE.TEXT 'TEXT)
                (\FPTYPE.BINARY
                  'BINARY)
                (\FPTYPE.UNKNOWN
                  NIL)
                (\TYPE.FROM.FILETYPE (fetch (M44FILEPROP
                  FPROPWORD0)
                    of BUF])
                  NIL)
                (SETQ BUF (\ADDBASE BUF (fetch (M44FILEPROP FPROPLENGTH) of BUF))
                (CREATIONDATE (\M44GETDATEPROP STREAM (INDEXF (fetch (\M44LeaderPage TimeCreate) of T)
                  T))
                (WRITEDATE (\M44GETDATEPROP STREAM (INDEXF (fetch (\M44LeaderPage TimeWrite) of T)
                  T))
                (READDATE (\M44GETDATEPROP STREAM (INDEXF (fetch (\M44LeaderPage TimeRead) of T)
                  T))
                (ICREATIONDATE
                  (\M44GETDATEPROP STREAM (INDEXF (fetch (\M44LeaderPage TimeCreate) of T))))
                (IWRITEDATE (\M44GETDATEPROP STREAM (INDEXF (fetch (\M44LeaderPage TimeWrite) of T))))
                (IREADDATE (\M44GETDATEPROP STREAM (INDEXF (fetch (\M44LeaderPage TimeRead) of T))))
                NIL])

```

(\M44GETDATEPROP

```

[LAMBDA (STREAM OFFSET STRINGIFY)
  (* bvm%: "27-May-84 22:57")
  ;; Returns the create/write/read date of STREAM that lives at OFFSET in its leader page, as a string if STRINGIFY is true, else as a Lisp date fixp
  (PROG ((DATABASE (\ADDBASE (\M44ReadLeaderPage STREAM)
    OFFSET))
    DAT)
    (SETQ DAT (\MAKENUMBER (\GETBASE DATABASE 0)
      (\GETBASE DATABASE 1)))
    (RETURN (COND
      ((NEQ DAT 0)
        (SETQ DAT (ALTO.TO.LISP.DATE DAT))
        (COND
          (STRINGIFY (GDATE DAT))
          (T DAT]))

```

(\M44GetFileName

```

[LAMBDA (NAME RECOG FDEV)
  (LET ((FS (\M44PARSEFILENAME NAME RECOG FDEV))
    DP UNAME)
    (AND FS (SETQ UNAME (fetch (FILESPEC UNAME) of FS))
      (\M44PACKFILENAME UNAME (SETQ DP (fetch (FILESPEC FSDIRPTR) of FS))
        (AND DP (fetch (M44DEVICE SYSDIROFD) of FDEV]))
    ; Edited 21-Jan-91 23:48 by jds

```

(\M44GetPageLoc

```

[LAMBDA (STREAM PAGENO CREATE?)
  ;; Look in the file's page map to find the disk address of the page. If the map does not include the page, then extend it appropriately. If page does
  ;; not exit, create it if CREATE? is true, else return \EOFDA
  (COND
    ((ILEQ PAGENO (fetch (M44STREAM LastPage) of STREAM))
      (COND
        ((IGREATERP PAGENO (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM))
          (\M44FillInMap STREAM PAGENO)))
        (\WORDELT (fetch (M44STREAM FILEPAGEMAP) of STREAM)
          (IPLUS PAGENO 2)))
      (CREATE? (\M44AddDiskPages STREAM PAGENO 0)
        (\M44GetPageLoc STREAM PAGENO))
      (T \EOFDA])
    ; Edited 21-Jan-91 23:35 by jds

```

(\M44KillFilePageMap

```

[LAMBDA (fHandle)
  (replace (M44STREAM FILEPAGEMAP) of fHandle with NIL)
  (replace (M44STREAM LASTMAPPEDPAGE) of fHandle with -1])
  ; Edited 21-Jan-91 23:35 by jds
  ; Remove the file's page map.

```

(M44MAKEDIRENTRY

[LAMBDA (FID UNAME NC FDEV) ; Edited 21-Jan-91 23:38 by jds

:: Makes a directory entry for a new file. FID is file's ID, NC the number of characters in the full Alto name.

```
(PROG ((DIRSTREAM (fetch (M44DEVICE SYSDIROFD) of FDEV))
      (VERSION (fetch (UNAME VERSION) of UNAME))
      POS)
      (SETQ POS (\FINDIRHOLE (LRSH (IPLUS NC 14)
                                   1)
                            DIRSTREAM))
      (\BOUNTS DIRSTREAM (fetch (FID FIDBLOCK) of FID)
        0
        (UNFOLD 5 BYTESPERWORD))
      (\BOUT DIRSTREAM NC)
```

:: Now write out the alto-style name 'name[.ext]!ver.' with ver omitted if 1; This is basically the same logic as is used to write the name in the leader page in M44CREATEFILE. We can't share cause here we do bouts, cause we might run over a page; there we must do PUTBASEBYTE's cause we can't set the fileptr to the leader page.

```
(for C in (fetch (UNAME ORIGCHARS) of UNAME) do (\BOUT DIRSTREAM C))
[COND
  ((NEQ VERSION 1)
   (\BOUT DIRSTREAM (CHARCODE !)))
  (LET ((*PRINT-BASE* 10))
   (PRIN3 VERSION DIRSTREAM))
  (\BOUT DIRSTREAM (CHARCODE %.)
  (COND
   ((EVENP NC BYTESPERWORD)
    (\BOUT DIRSTREAM 0)))
  (\SETFILEPTR DIRSTREAM POS)
  (\BOUT DIRSTREAM (LOGOR 4 (\PEEKBIN DIRSTREAM)))
  (FORCEOUTPUT DIRSTREAM)
  (RETURN POS))
```

; When everything is ready, finally change the type from hole to file.

(M44OpenFile

[LAMBDA (NAME ACCESS RECOG PARAMETERS FDEV OLDSTREAM) ; Edited 21-Jan-91 23:38 by jds

:: Open a Model44 file. Gets the physical end of file and sets up ofd

```
(PROG (PAGESTIMATE STREAM CRDATE TYPE DON'T.CHANGE.DATE X) ; if file is open in a conflicting way, barf
      [COND
        ((NEQ ACCESS 'INPUT) ; Interesting parameters when creating a file
         (for X in PARAMETERS do (SELECTQ (CAR (LISTP X))
                                           (LENGTH (SETQ PAGESTIMATE (IPLUS 2 (FOLDHI (CADR X)
                                                                                       BYTESPERPAGE))))
                                           (CREATIONDATE (SETQ CRDATE (IDATE (CADR X))))
                                           (ICREATIONDATE
                                            (SETQ CRDATE (CADR X)))
                                           (TYPE (SETQ TYPE (CADR X)))
                                           (DON'T.CHANGE.DATE
                                            (SETQ DON'T.CHANGE.DATE T))
                                           NIL]
```

```
(COND
 [ (type? STREAM NAME)
  (COND
```

```
((OR (fetch (M44DEVICE DSKPASSWORDOK) of (fetch DEVICE of NAME))
      (EQ (fetch (FID W0) of (fetch (M44STREAM FID) of NAME))
          32768))
  (M44CompleteFH (SETQ STREAM NAME)))
(T (RETURN))
([NULL (SETQ STREAM (\M44GetFileHandle NAME RECOG FDEV NIL (NEQ ACCESS 'INPUT))
  (RETURN NIL))
  ; Make sure password is ok if trying to reopen anything but a
  ; directory
  ; File not found. Return NIL to let generic open generate a FILE
  ; NOT FOUND error
```

```
(if OLDSTREAM
 then ; REOPENFILE--nothing more to do
 (RETURN STREAM))
```

```
[COND
 ([AND PAGESTIMATE (IGREATERP PAGESTIMATE (IPLUS (fetch (M44DEVICE DISKFREEPAGES) of FDEV)
                                                    (COND
              ((type? STREAM STREAM)
               (fetch (M44STREAM LastPage) of STREAM))
              (T
               ; New file
               0)
              (RETURN (LISPERROR "FILE SYSTEM RESOURCES EXCEEDED" (COND
                ((type? STREAM STREAM)
                 (fetch FULLFILENAME of STREAM))
                (T NAME)]
```

```
(COND
 (CRDATE ; Convert to alto format
  (COND
   ([NOT (type? FIXP (SETQ CRDATE (LISP.TO.ALTO.DATE CRDATE))
            ; sigh, wanted a number box
            (\PUTBASEFIXP (SETQ X (create FIXP))
                          0 CRDATE)
```

```

      (SETQ CRDATE X)
[COND
  ((NOT (type? STREAM STREAM)) ; New file
   (SETQ STREAM (\M44CREATEFILE FDEV STREAM PAGESTIMATE CRDATE TYPE)))
  (T ; Old file
   [LET ((MYNAME (fetch FULLFILENAME of STREAM)))
      (COND
        ([for OTHER in (fetch (FDEV OPENFILELST) of FDEV) when (STRING-EQUAL (fetch FULLFILENAME of OTHER) MYNAME)
         do (RETURN (OR (NEQ ACCESS 'INPUT)
                       (NEQ (fetch ACCESS of OTHER) 'INPUT)
                       ; Access conflict with existing open file
                       (RETURN (LISPERROR "FILE WON'T OPEN" MYNAME)
                                ; File is EMPTY even if it is old
                                (replace EPAGE of STREAM with (replace EOFFSET of STREAM with 0)
                                ; Leader page is read in during STREAM initialization
                                (COND
                                  ((EQ ACCESS 'OUTPUT)
                                   (NOT DON'T.CHANGE.DATE)
                                   (\M44SetAccessTimes STREAM ACCESS CRDATE) ; Resets validation
                                   (\M44WriteLeaderPage STREAM) ; We write out accumulated changes to leader page
                                  ]
                                (COND
                                  (CRDATE (replace NONDEFAULTDATEFLG of STREAM with T)))
                                (RETURN STREAM]))
      ]
  (COND
    (CRDATE (replace NONDEFAULTDATEFLG of STREAM with T)))
  (RETURN STREAM))

```

(\M44OPENFILEFROMFP

```

[LAMBDA (DEV NAME ACCESS FID DIRINFO) ; Edited 21-Jan-91 23:36 by jds
                                           ; Opens a disk file given its FP
  (LET ((STREAM (create M44STREAM)))
      (replace FULLFILENAME of STREAM with (SETQ NAME (PACK* '{ (fetch (FDEV DEVICENAME) of DEV)
                                                                '} NAME)))
      (replace DEVICE of STREAM with DEV)
      (replace (M44STREAM FID) of STREAM with FID)
      (replace (M44STREAM DIRINFO) of STREAM with DIRINFO)
      (replace MULTIBUFFERHINT of STREAM with \M44MULTIFLG)
      (\OPENFILE STREAM ACCESS)
      (replace USERVISIBLE of STREAM with NIL)
      STREAM])

```

(\M44ReadDiskPage

```

[LAMBDA (STREAM PAGENO BUF) ; Edited 21-Jan-91 23:42 by jds
  ;; The functions for reading a disk page called by \M44ReadPages. Returns the number of bytes read. If PAGEADDR is 0, then assume 0 bytes
  ;; read. Fill the BUF with zeros beyond the last byte read.
  (COND
    ((AND (IGEQ PAGENO (fetch EPAGE of STREAM))
          (OR (NOT (IEQP PAGENO (fetch EPAGE of STREAM)))
              (EQ (fetch EOFFSET of STREAM) 0))) ; Asking for page after eof. PMAP system really ought to catch
          ; this itself
     (\CLEARWORDS BUF WORDSPERPAGE)
     0)
    (T (PROG ((PAGEADDR (\M44GetPageLoc STREAM PAGENO))
              (BFSPG# (ADD1 PAGENO)))
        (RETURN (COND
                  ((EQ PAGEADDR \EOFDA) ; no bytes read, fill with zeroes.
                   (\CLEARWORDS BUF WORDSPERPAGE)
                   0)
                  ((EQ PAGEADDR \FILLINDA)
                   (SHOULDNT))
                  ((EQ (\ACTONDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
                                           BUF
                                           (fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP) of STREAM))
                                           -1 STREAM BFSPG# BFSPG# \DC.READD)
                                           BFSPG#)
                   BYTESPERPAGE)
                  (T ;; if READDISKPAGE returns NIL, presumably there is an error of some kind, hope it was with the file map and
                     ;; try again.
                     (\M44KillFilePageMap STREAM)
                     (\M44ReadDiskPage STREAM PAGENO BUF]))
        ))

```

(\M44ReadLeaderPage

```

[LAMBDA (STREAM AGAIN) ; Edited 21-Jan-91 23:42 by jds
  ;; Returns the leader page of STREAM, reading it if necessary. If AGAIN is true, will read it afresh even if it already has a cached leader page
  ;; File leader page format: Words 0-1, time created. Words 2-3, time last written. Words 4-5, time last read. Words 6-25, name of file. Words
  ;; 26-235, leader properties. Words 236-245, spare. Word 246, property pointer. Word 247, change serial number. Words 248-252, STREAM hint
  ;; for directory. Word 253, disk address of last page. Word 254, page number of last page. Word 255, number of bytes on last page.
  (PROG ((BUFFER (fetch (M44STREAM LEADERPAGE) of STREAM))

```



```
(COND
  [(NULL BUFFER)
   (SETQ BUFFER (NCREATE 'VMEMPAGEP)
   (NOT AGAIN)
   (RETURN BUFFER)))
 (\ACTONDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
  BUFFER
  (fetch (ARRAYP BASE) of (OR (fetch (M44STREAM FILEPAGEMAP) of STREAM)
  (MAKELEADERDAS STREAM)))
  -1 STREAM 0 0 \DC.READD)
 (replace (M44STREAM LEADERPAGE) of STREAM with BUFFER)
 (RETURN BUFFER])
```

(M44ReadPages

```
[LAMBDA (STREAM FIRSTPAGE# BUFFERS)
  (* bvm%: "26-DEC-81 23:50")
  ; Read pages from a Model44 file.
  (for BUF inside BUFFERS as PAGENO from FIRSTPAGE# sum (M44ReadDiskPage STREAM PAGENO BUF])
```

(M44SetAccessTimes

```
[LAMBDA (STREAM ACCESS CRDATE) ; Edited 21-Jan-91 23:36 by jds
```

;;; Set the 'last read' and/or 'last written' times in the leader page according to access, which is assumed to be either INPUT, OUTPUT, BOTH, or
;;; APPEND.

```
(PROG ((DAT (\DAYTIME0 (create FIXP)))
  (BUF (fetch (M44STREAM LEADERPAGE) of STREAM)))
  ;; Note: DAYTIME0 returns an Alto time, not Lisp time. This is consistent with the dates in the leader page
  (SELECTQ ACCESS
    ((OUTPUT BOTH APPEND)
     (\BLT (LOCF (fetch (\M44LeaderPage TimeCreate) of BUF))
      (OR CRDATE DAT)
      WORDSPERCELL)
     (\BLT (LOCF (fetch (\M44LeaderPage TimeWrite) of BUF))
      DAT WORDSPERCELL)
     ; Must revalidate because write DAT has changed
     (UPDATEVALIDATION STREAM BUF))
    NIL)
  (SELECTQ ACCESS
    ((INPUT BOTH)
     (\BLT (LOCF (fetch (\M44LeaderPage TimeRead) of BUF))
      DAT WORDSPERCELL))
    NIL))
```

(M44SetEndOfFile

```
[LAMBDA (STREAM EPAGE EOFFSET UPDATENOW) ; Edited 21-Jan-91 23:36 by jds
```

;;; Reset the file's leader page end-of-file hint. If UPDATENOW is NIL, then simply update the leader page. If it is not, then read and write the
;;; leader page.

```
(UNINTERRUPTABLY
  ;; Must update STREAM handle and leader page in synch
  (replace (M44STREAM LastPage) of STREAM with EPAGE)
  (replace (M44STREAM LastOffset) of STREAM with EOFFSET)
  [LET ((LEADERPAGE (M44ReadLeaderPage STREAM))
    (if (NEQ (fetch (\M44LeaderPage LastPageNumber) of LEADERPAGE)
      (ADD1 EPAGE))
    then
    ; if LastPage hasn't changed, don't do anything
    ; ADD1 because M44 counts from 1
    (replace (\M44LeaderPage LastPageAddress) of LEADERPAGE with (M44GetPageLoc STREAM EPAGE))
    (replace (\M44LeaderPage LastPageNumber) of LEADERPAGE with (ADD1 EPAGE)))
  (replace (\M44LeaderPage LastPageByteCount) of LEADERPAGE with EOFFSET)
  (COND
    (UPDATENOW (M44WriteLeaderPage STREAM]))]
```

(M44SetFileInfo

```
[LAMBDA (STREAM ATTRIBUTE VALUE DEV) ; Edited 21-Jan-91 23:34 by jds
```

```
(PROG ((WASOPEN (type? STREAM STREAM))
  (SELECTQ ATTRIBUTE
    (CREATIONDATE (SETQ VALUE (OR (IDATE VALUE)
    (LISPERROR "ILLEGAL ARG" VALUE))))
    (ICREATIONDATE
     (OR (FIXP VALUE)
     (LISPERROR "NON-NUMERIC ARG" VALUE)))
    (TYPE)
    (RETURN))
  (RETURN (COND
    ((OR WASOPEN (SETQ STREAM (M44GetFileHandle STREAM 'OLD DEV T)))
     (COND
       ((SELECTQ ATTRIBUTE
         (TYPE (M44SETFILETYPE STREAM VALUE))
         (PROGN (replace (\M44LeaderPage TimeCreate) of (M44ReadLeaderPage STREAM)
           with (LISP.TO.ALTO.DATE VALUE))
         T))
```

(\M44WriteLeaderPage STREAM)
T])

(\M44SETFILETYPE

[LAMBDA (STREAM TYPE) ; Edited 21-Jan-91 23:44 by jds

:: Set TYPE attribute of file to be TYPE -- assumes someone else will be writing out the leader page later

(PROG ((TYPECODE (SELECTQ TYPE
(TEXT \FPTYPE.TEXT)
(BINARY \FPTYPE.BINARY)
(NIL \FPTYPE.UNKNOWN)
(OR (\FILETYPE.FROM.TYPE TYPE)
\FPTYPE.BINARY)))
(BUF (\M44ReadLeaderPage STREAM))
PTR TOTALLENGTH)

:: Computation of TYPECODE done this way for backward compatibility -- the \FPTYPE.xx constants were defined before
:: \FILETYPE.FROM.TYPE was written, and the numbers are incompatible

(SETQ PTR (\ADDBASE BUF (fetch (\M44LeaderPage PropertyBegin) of BUF)))
(SETQ TOTALLENGTH (fetch (\M44LeaderPage PropertyLength) of BUF))
(RETURN (while (IGREATERP TOTALLENGTH 0)
do (SELECTC (fetch (M44FILEPROP FPROPTYPE) of PTR)
(0 ; End of properties
(RETURN (COND
((IGREATERP TOTALLENGTH 1)
(replace (M44FILEPROP FPROPWORD0) of PTR with TYPECODE)
(replace (M44FILEPROP FPROPLENGTH) of PTR with 2)
(replace (M44FILEPROP FPROPTYPE) of PTR with \FPROP.TYPE)
T))))
(\FPROP.TYPE ; Already has a type, change it
(replace (M44FILEPROP FPROPWORD0) of PTR with TYPECODE)
(RETURN T))
NIL)
(SETQ PTR (\ADDBASE PTR (fetch (M44FILEPROP FPROPLENGTH) of PTR)))
(SETQ TOTALLENGTH (IDIFFERENCE TOTALLENGTH (fetch (M44FILEPROP FPROPLENGTH) of PTR))

(\M44TruncateFile

[LAMBDA (STREAM LP LO UPDATENOW) ; Edited 21-Jan-91 23:36 by jds

:: Resets the length of the file to LP page and LO offset. Can both shorten and lengthen files.

[COND
((NOT LP)
(SETQ LP (fetch EPAGE of STREAM))
(SETQ LO (fetch EOFSET of STREAM])
(COND
((IGREATERP LP (fetch (M44STREAM LastPage) of STREAM))
(\M44AddDiskPages STREAM LP LO))
((ILESSP LP (fetch (M44STREAM LastPage) of STREAM))
(\M44DELETEPAGES STREAM (ADD1 LP))
(COND
((ILESSP LP (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM))
(for I from (ADD1 LP) to (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM)
do (SETA (fetch (M44STREAM FILEPAGEMAP) of STREAM)
(IPLUS I 2)
\EOFDA))
(replace (M44STREAM LASTMAPPEDPAGE) of STREAM with LP)))
(\M44SetEndOfFile STREAM LP LO) ; Now need to rewrite last page with new length, null next pointer
(\MAPPAGE LP STREAM)
(\SETIODIRTY STREAM LP)
(FORCEOUTPUT STREAM))
(T (replace (M44STREAM LastOffset) of STREAM with LO)))
(AND UPDATENOW (\M44SetEndOfFile STREAM LP LO T))
STREAM])

(\M44WriteDiskPage

[LAMBDA (STREAM PAGENO BUF NBYTES) ; Edited 21-Jan-91 23:42 by jds
; Write a disk page on the Model44.
; Ensure that PAGENO is in map

(\M44GetPageLoc STREAM PAGENO T)
(PROG ((BFSPG# (ADD1 PAGENO)))
(RETURN (COND
([COND
((NEQ PAGENO (fetch (M44STREAM LastPage) of STREAM))
; Writing only data
(\ACTONDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
BUF
(fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP) of STREAM))
-1 STREAM BFSPG# BFSPG# \DC.WRITED))
(T ; When writing last page, need to fill in the numchars field of
; label, so this is harder
(COND
((EQ PAGENO (fetch EPAGE of STREAM))
(EQ (\WRITEDISKPAGE (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
BUF
(fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP) of STREAM))

```

-1 STREAM BFSPG# BFSPG# NIL NIL NBYTES)
BFSPG#))
(T ;; We will have to write more pages after this one, too, unless the file is truncated back to here, so extend
;; the file while we're at it. This may save a call to \ADDDISKPAGES
[COND
((ILEQ (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM)
PAGENO)
(\M44ExtendFilePageMap STREAM (ADD1 PAGENO)]
(COND
((EQ (\WRITEDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
(LIST BUF NIL)
(fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP)
of STREAM))
-1 STREAM BFSPG# (ADD1 BFSPG#)
NIL NIL 0)
(ADD1 BFSPG#)) ; Write two pages, the second of which is blank
(replace (M44STREAM LastPage) of STREAM with (ADD1 PAGENO))
(replace (M44STREAM LastOffset) of STREAM with 0)
T]
NBYTES)
(T (\M44KillFilePageMap STREAM)
(\M44WriteDiskPage STREAM PAGENO BUF NBYTES])

```

(\M44WriteLeaderPage

```

[LAMBDA (STREAM) ; Edited 21-Jan-91 23:42 by jds
; Write the file's leader page
(PROG ((BUFFER (fetch (M44STREAM LEADERPAGE) of STREAM)))
(AND BUFFER (\ACTONDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
BUFFER
(fetch (ARRAYP BASE) of (OR (fetch (M44STREAM FILEPAGEMAP) of STREAM)
(\MAKELEADERDAS STREAM)))
-1 STREAM 0 0 \DC.WRITED])

```

(\M44WritePages

```

[LAMBDA (STREAM FIRSTPAGE# BUFFERS) ; Edited 21-Jan-91 23:36 by jds
; Write pages onto a Model44 file.
(PROG ([NPAGES (COND
((NLISTP BUFFERS)
1)
(T (for B in BUFFERS sum 1]
LASTPAGE#)
(COND
((fetch REVALIDATEFLG of STREAM)
;; Need to update creationdate, since a SAVEVM etc has occurred since the last write. Otherwise, it is possible to see a change to
;; the file but no change to the creationdate
(\M44SetAccessTimes STREAM 'OUTPUT)
(\M44WriteLeaderPage STREAM)
(replace REVALIDATEFLG of STREAM with NIL)))
(\M44GetPageLoc STREAM FIRSTPAGE# T) ; Make sure we know where we are starting to write
(COND
([ILESSP (fetch (M44STREAM LASTMAPPEDPAGE) of STREAM)
(SETQ LASTPAGE# (IPLUS FIRSTPAGE# (SUB1 NPAGES)
; Need enough pagemap to cover everything we write
(\M44ExtendFilePageMap STREAM (ADD1 LASTPAGE#)
(COND
([AND (IGEQ NPAGES \#DISKBUFFERS)
(for B in BUFFERS thereis (NOT (EMADDRESSP B)
;; More pages to write than we have disk buffers to do it in one command, so break it up. Buffers already in emulator space are free,
;; though, so we can write lots of them
(bind (MAXPAGES _ (SUB1 \#DISKBUFFERS)) do (\M44WritePages1 STREAM FIRSTPAGE# (IPLUS FIRSTPAGE#
(SUB1 MAXPAGES)
)
(to MAXPAGES collect (pop BUFFERS)))
(add FIRSTPAGE# MAXPAGES)
(SETQ NPAGES (IDIFFERENCE NPAGES MAXPAGES))
repeatwhile (IGREATERP NPAGES MAXPAGES]
(\M44WritePages1 STREAM FIRSTPAGE# LASTPAGE# BUFFERS])

```

(\M44WritePages1

```

[LAMBDA (STREAM FIRSTPAGE# LASTPAGE# BUFFERS) ; Edited 21-Jan-91 23:42 by jds

```

;;; Writes BUFFERS to STREAM, covering pages FIRSTPAGE# thru LASTPAGE#. Caller guarantees that we have enough disk buffers to do it. ---
 ;;; There are two cases: easy one is if the pages already exist, in which case we just rewrite their data; hard case is writing pages at end of file, in which
 ;;; case we need to write labels and maybe allocate pages

```

(COND
((ILESSP LASTPAGE# (fetch (M44STREAM LastPage) of STREAM)) ; Writing only data
(\ACTONDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
BUFFERS
(fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP) of STREAM))
-1 STREAM (.LISP.TO.BFS. FIRSTPAGE#)

```

```

      (.LISP.TO.BFS. LASTPAGE#)
      \DC.WRITED)
(T
; When writing last page, need to fill in the numchars field of
; label, so this is harder
(PROG (MYBUFS NBYTES)
 [SETQ MYBUFS (COND
 ((AND (EQ LASTPAGE# (fetch EPAGE of STREAM))
 (NEQ (SETQ NBYTES (fetch EOFFSET of STREAM))
 BYTESPERPAGE)) ; Only write to the end of the file
 BUFFERS)
 (T ;; We will have to write more pages after this one, too, unless the file is truncated back to here, so extend
 ;; the file while we're at it. This may save a call to \ADDDISKPAGES
 (PROG1 (SETQ MYBUFS (CONS))
 [for B inside BUFFERS do (RPLACA MYBUFS B)
 (SETQ MYBUFS (CDR (RPLACD MYBUFS (CONS)
 (RPLACD (RPLACA MYBUFS NIL)
 NIL) ; Write a final blank page
 (SETQ NBYTES 0)
 (add LASTPAGE# 1))])
 (\WRITEDISKPAGES (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM))
 MYBUFS
 (fetch (ARRAYP BASE) of (fetch (M44STREAM FILEPAGEMAP) of STREAM))
 -1 STREAM (.LISP.TO.BFS. FIRSTPAGE#)
 (.LISP.TO.BFS. LASTPAGE#)
 NIL NIL NBYTES)
 (replace (M44STREAM LastPage) of STREAM with LASTPAGE#)
 (replace (M44STREAM LastOffset) of STREAM with NBYTES])
)
)

```

:: Disk allocation

(DEFINEQ

(ADDDISKPAGES

```

[LAMBDA (STREAM FIRSTNEWPAGE NPAGES DAs LASTNUMCHARS) ; Edited 21-Jan-91 23:42 by jds
;; Adds to file STREAM NPAGES, where FIRSTNEWPAGE-1 is the last existing page. DAs is the vector of disk addresses, where first element
;; corresponds to BFS page -1 ; Note FIRSTNEWPAGE is in Lisp terms, so it is actually
; LASTOLDPAGE for the BFS
(PROG ((LASTPAGEBUF (NCREATE 'VMEMPAGEP))
 (LASTEXISTINGPAGE FIRSTNEWPAGE)
 (DSK (fetch (M44DEVICE DSKOBJ) of (fetch DEVICE of STREAM)))
 BUFFERS CHUNK)
 (SETQ BUFFERS (CONS LASTPAGEBUF (for I from 1 to (IMIN NPAGES \MAXDISKDA$ collect NIL)))
 (\ACTONDISKPAGES DSK LASTPAGEBUF DAs -1 STREAM LASTEXISTINGPAGE LASTEXISTINGPAGE \DC.READD NIL NIL NIL
 LASTEXISTINGPAGE) ; Read last existing page, so we can rewrite it with new label
 (while (IGREATERP NPAGES 0) do (SETQ CHUNK (IMIN \MAXDISKDA$ NPAGES))
 (\WRITEDISKPAGES DSK BUFFERS DAs -1 STREAM LASTEXISTINGPAGE
 (IPLUS LASTEXISTINGPAGE CHUNK)
 NIL NIL LASTNUMCHARS LASTEXISTINGPAGE)
 (RPLACA BUFFERS NIL)
 (add LASTEXISTINGPAGE CHUNK)
 (SETQ NPAGES (IDIFFERENCE NPAGES CHUNK))
)
)

```

(M44DELETEPAGES

```

[LAMBDA (STREAM FIRSTPAGE) ; Edited 21-Jan-91 23:42 by jds
; FIRSTPAGE is in Lisp terms, i.e. -1 = leader page
(PROG ((DEV (fetch DEVICE of STREAM))
 (NPAGES (COND
 ((fetch VALIDATION of STREAM)
 (IPLUS (ADD1 (IDIFFERENCE (fetch (M44STREAM LastPage) of STREAM)
 FIRSTPAGE))
 2))
 (T PageMapIncrement)))
 (PN (ADD1 FIRSTPAGE))
 DAs FIRSTDA LASTPAGESEEN DSK)
;; NPAGES is used to decide how much to do at once. Need be no more than number of pages known to exist. The ADD1 is that, plus two for
;; the pages around it
(COND
 ((ILESSP NPAGES 2) ; Nothing to delete
 (RETURN)))
 (SETQ DSK (fetch (M44DEVICE DSKOBJ) of DEV)) (* (\FLUSHDISKDESCRIPTOR (EMPOINTER
 (fetch (DSKOBJ DSKDDMGR) of DSK))
 (fetch (DSKOBJ ALTODSKOBJ) of DSK)))
 ; Tell Alto to clear out anything it knows about dd
 ; IF STREAM:LASTMAPPEDPAGE GE FIRSTPAGE+NPAGES
 ; THEN DAs _ STREAM:FILEPAGEMAP DAorigin _ -1
 (SETQ DAs (ARRAY (SETQ NPAGES (IMIN NPAGES \MAXDISKDA$)
 'WORD NIL 0))
 [SETQ FIRSTDA (COND
 [(EQ FIRSTPAGE -1)
 (fetch (FP FPLEADERVDA) of (fetch (FID FIDBLOCK) of (fetch (M44STREAM FID) of STREAM))]
 (T (M44GetPageLoc STREAM FIRSTPAGE)
)
)
)

```

```

(while (NEQ FIRSTDA \EOFDA) do (SETA DAs 0 \FILLINDA)
      (SETA DAs 1 FIRSTDA) ; Corresponds to PN
      (for I from 2 to (SUB1 NPAGES) do (SETA DAs I \FILLINDA))
      [SETQ LASTPAGESEEN (\ACTONDISKPAGES DSK NIL (fetch (ARRAYP BASE)
                                                         of DAs)
                        (SUB1 PN)
                        STREAM PN (IPLUS PN NPAGES -3)
                        \DC.READD NIL NIL NIL
                        (ADD1 (fetch EPAGE of STREAM)
                             ; Read DAs for the next NPAGES-2
                             \WRITEDISKPAGE DSK NIL (fetch (ARRAYP BASE) of DAs)
                        (SUB1 PN)
                        \FREEPAGEFID PN LASTPAGESEEN (UNSIGNED -1 BITSPERWORD))
      [for I from PN to LASTPAGESEEN
        do (M44MARKPAGEFREE DEV (ELT DAs (ADD1 (IDIFFERENCE I PN)
                                                (SETQ FIRSTDA (ELT DAs (IPLUS (IDIFFERENCE LASTPAGESEEN PN)
                                                                    2)))
                                                (SETQ PN (ADD1 LASTPAGESEEN))))
          (* (FLUSHMAP (fetch (M44DEVICE DISKDESCRIPTOROFD)
                              of DEV)) (FORGETPAGES (fetch
                                                    (M44DEVICE DISKDESCRIPTOROFD) of DEV)))
      (\M44FLUSHDISKDESCRIPTOR DEV])

```

(ASSIGNDISKPAGE

[LAMBDA (DSK PREVDA)

; Edited 21-Jan-91 23:32 by jds

::: Assigns a new page on DSK. If PREVDA is \EOFDA will pick random page, otherwise will attempt to allocate PREVDA+1. Returns NIL if disk is full

```

(PROG ([VDA (COND
      ((OR (EQ PREVDA \EOFDA)
           (COND
            ((EQ PREVDA \FILLINDA)
             (AND \DISKDEBUG (RAID "[Disk debug] \ASSIGNDISKPAGE called with \FILLINDA. ^N to
                                continue"))
            T)))
      (fetch (DSKOBJ DISKLASTPAGEALLOC) of DSK))
      (T (ADD1 PREVDA])
      (DD (fetch (DSKOBJ DISKDESCRIPTOROFD) of DSK))
      (MASK 128)
      BITS A LOOPEDONCE FREE)
      (OR (fetch (DSKOBJ DDVALID) of DSK)
          (RAID "DISKDESCRIPTOR not open" DSK))
      (\SETFILEPTR DD (IPLUS \DDBITTABSTART (FOLDLO VDA BITSPERBYTE)))
      (SETQ A (MOD VDA BITSPERBYTE))
      (FRPTQ A (SETQ MASK (LRSH MASK 1)))
LP (COND
    ((NULL (SETQ BITS (\BIN DD)))
     ;; End of file -- wrap around to start. This technique takes longer than necessary to bomb out when disk is full, but who cares?
     (COND
      (LOOPEDONCE (RETURN NIL)))
      (SETQ LOOPEDONCE T)
      (\SETFILEPTR DD \DDBITTABSTART))
    ((NEQ BITS 255)
     (until (OR (EQ (LOGAND BITS MASK)
                    0)
                (EQ (SETQ MASK (LRSH MASK 1))
                    0)))
            do (add A 1))
    (COND
     ((NEQ MASK 0)
      (\BACKFILEPTR DD)
      (SETQ VDA (IPLUS (UNFOLD (IDIFFERENCE (\GETFILEPTR DD)
                                           \DDBITTABSTART)
                               BITSPERBYTE)
                       A))
      (\BOUT DD (LOGOR BITS MASK))
      ; Found a free page
      ; Set bit indicating we snarfed this page
      ; Decrement free page count hint
      [replace (DSKOBJ DISKFREEPAGES) of DSK with (COND
                                                    ((EQ (SETQ FREE (fetch (DSKOBJ DISKFREEPAGES)
                                                                    of DSK))
                                                         0)
             (AND \DISKDEBUG (RAID "[Disk debug] Free page
                                    hint went negative. ^N to
                                    continue"))
             (\COUNTDISKFREEPAGES DD))
            (T (SUB1 FREE])
      (replace (DSKOBJ DISKLASTPAGEALLOC) of DSK with VDA)
      (replace (DSKOBJ DDDIRTY) of DSK with T)
      (RETURN VDA)))
    (SETQ MASK 128)
    (SETQ A 0)))
(GO LP])

```

(COUNTDISKFREEPAGES

[LAMBDA (DD) (* bvm%: "13-Feb-85 19:32")

::: Counts number of free pages on a disk. DD is the diskdescriptor stream

```

[OR (type? STREAM DD)
  (SETQ DD (\OPENDISKDESCRIPTOR (\GETDEVICEFROMNAME (OR DD 'DSK)
  (PROG ((CNT 0)
    MASK BITS)
    (\SETFILEPTR DD \DDBITABSTART)
  LP [COND
    ((NULL (SETQ BITS (\BIN DD))) ; End of file
      (RETURN CNT))
    ((EQ BITS 0)
      (add CNT 8))
    ((NEQ BITS 255)
      (SETQ MASK 128)
      (do (COND
        ((EQ (LOGAND BITS MASK)
          0)
          (add CNT 1)))
        until (EQ (SETQ MASK (LRSH MASK 1))
          0)
      (GO LP])

```

(\M44MARKPAGEFREE

[LAMBDA (DEV DA) (* bvm%: "17-Jan-85 17:11")

::: Mark disk address DA on disk device DEV free

```

(PROG ((DSK (COND
  ((type? FDEV DEV)
  (fetch (M44DEVICE DSKOBJ) of DEV))
  (T DEV)))
  DD BITS MASK)
[SETQ DD (COND
  ((fetch (DSKOBJ DDVALID) of DSK)
  (fetch (DSKOBJ DISKDESCRIPTOROFD) of DSK))
  (T (\OPENDISKDESCRIPTOR DEV)
  (SETFILEPTR DD (IPLUS \DDBITABSTART (FOLDLO DA BITSPERBYTE)))
  (SETQ BITS (\BIN DD))
  [SETQ MASK (LLSH 1 (IDIFFERENCE 7 (MOD DA BITSPERBYTE)
  (COND
    ((NEQ (LOGAND BITS MASK)
      0) ; Page is marked occupied, so free it
      (\BACKFILEPTR DD)
      (\BOUT DD (LOGXOR BITS MASK))
      (add (fetch (DSKOBJ DISKFREEPAGES) of DSK)
        1)
      (replace (DSKOBJ DDDIRTY) of DSK with T))

```

(\M44FLUSHDISKDESCRIPTOR

[LAMBDA (DEV) ; Edited 21-Jan-91 23:32 by jds

```

(PROG ((DSK (COND
  ((type? FDEV DEV)
  (fetch (M44DEVICE DSKOBJ) of DEV))
  (T DEV)))
  DD)
[OR (fetch (DSKOBJ DDDIRTY) of DSK)
  (RETURN))
[OR (SETQ DD (fetch (DSKOBJ DISKDESCRIPTOROFD) of DSK))
  (RETURN (RAID "[Disk debug] no disk descriptor stream"))
  (\SETFILEPTR DD \OFFSET.DISKLASTSERIAL#)
  (\BOUTS DD (LOC (fetch (DSKOBJ DISKLASTSERIAL#) of DSK))
    0 \NBYTES.DISKINFO) ; Copy interesting stuff into diskdescriptor header
  (FORCEOUTPUT DD)
  (replace (DSKOBJ DDDIRTY) of DSK with NIL)
  (RETURN T))

```

(\MAKELEADERDAS

[LAMBDA (STREAM) ; Edited 21-Jan-91 23:30 by jds

::: Makes a page map for STREAM that includes the leader vda

```

(PROG ((MAP (ARRAY 4 'WORD \FILLINDA 0)))
  (SETA MAP 0 \EOFDA)
  [SETA MAP 1 (fetch (FP PLEADERVDA) of (fetch (ARRAY BASE) of (fetch (M44STREAM FID) of STREAM)
  (replace (M44STREAM FILEPAGEMAP) of STREAM with MAP)
  (replace (M44STREAM LASTMAPPEDPAGE) of STREAM with -1)
  (RETURN MAP])

```

(\DISKFREEPAGES

[LAMBDA (DSK RECOMPUTE) (* ejs%: " 7-Nov-85 16:33")

```

(SELECTC \MACHINETYPE
  ((LIST \DANDELION \DAYBREAK) ; Temporary until this become a device op

```

```
(\DFSFreeDiskPages DSK RECOMPUTE))
(\M44FREEPAGECOUNT [COND
  ((type? FDEV DSK)
   DSK)
  (T (\GETDEVICEFROMNAME (OR DSK 'DSK]
    NIL RECOMPUTE]))
```

```
(\M44FREEPAGECOUNT
 [LAMBDA (DEV DIRECTORY RECOMPUTE) (* bvm%: "12-Oct-85 15:43")
 (PROG (CNT)
 (COND
 ((NOT (type? M44DEVICE DEV))
 (\ILLEGAL.ARG DEV))
 (RETURN (COND
 (RECOMPUTE (SETQ CNT (\COUNTDISKFREEPAGES (\OPENDISKDESCRIPTOR DEV)))
 (COND
 ((NEQ CNT (fetch (M44DEVICE DISKFREEPAGES) of DEV))
 (replace (M44DEVICE DISKFREEPAGES) of DEV with CNT)
 (replace (M44DEVICE DDDIRTY) of DEV with T)))
 CNT)
 (T (fetch (M44DEVICE DISKFREEPAGES) of DEV]))
 )
```

```
(RPAQ? \M44MULTFLG T)
```

```
(DECLARE%: DONTCOPY
```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS UCASECHAR MACRO [(C)
 (COND
 ((ILESSP C (CHARCODE a))
 C)
 (T (IDIFFERENCE C (IDIFFERENCE (CHARCODE a)
 (CHARCODE A))
```

```
(PUTPROPS UPDATEVALIDATION MACRO [(STREAM BUF)
 (replace VALIDATION of STREAM with (\MAKENUMBER (\GETBASE BUF 1)
 (\GETBASE BUF 3])
```

```
(DECLARE%: EVAL@COMPILE
```

```
[ACCESSFNS M44DEVICE ((DSKOBJ (fetch DEVICEINFO of DATUM)
 (replace DEVICEINFO of DATUM with NEWVALUE))
 (TYPE? (AND (type? FDEV DATUM)
 (EQ (fetch OPENFILE of DATUM)
 '\M44OpenFile]
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS \M44MULTFLG \DISKNAMECASEARRAY)
```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS .LISP.TO.BFS. MACRO (= . ADD1))
```

```
(PUTPROPS .BFS.TO.LISP. MACRO (= . SUB1))
```

```
(PUTPROPS .DISKCASEARRAY. MACRO [NIL (fetch (ARRAYP BASE) of (\DTEST \DISKNAMECASEARRAY 'ARRAYP])
```

```
(DECLARE%: EVAL@COMPILE
```

```
(RPAQQ PageMapIncrement 64)
```

```
(RPAQQ \MAX.ALTO.NAME.LENGTH 39)
```

```
(CONSTANTS (PageMapIncrement 64)
 (\MAX.ALTO.NAME.LENGTH 39))
```

:: File properties

```
(DECLARE%: EVAL@COMPILE
```

```
(BLOCKRECORD M44FILEPROP ((FPROPTYPE BYTE) ; Type of property
 (FPROPLENGTH BYTE) ; Length of entire entry in words
 (FPROPWORD0 WORD) ; value starts here
```

; Overlays a piece of leader page to describe a file property

)

)

(RPAQQ **FPROPTYPES** ((\FPROP.TYPE 136)
(\FPROP.PAGEMAP 137)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \FPROP.TYPE 136)

(RPAQQ \FPROP.PAGEMAP 137)

(CONSTANTS (\FPROP.TYPE 136)
(\FPROP.PAGEMAP 137))

(RPAQQ **FPTYPES** ((\FPTYPE.UNKNOWN 0)
(\FPTYPE.TEXT 1)
(\FPTYPE.BINARY 2)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \FPTYPE.UNKNOWN 0)

(RPAQQ \FPTYPE.TEXT 1)

(RPAQQ \FPTYPE.BINARY 2)

(CONSTANTS (\FPTYPE.UNKNOWN 0)
(\FPTYPE.TEXT 1)
(\FPTYPE.BINARY 2))

(DECLARE%: DONTCOPY
DECLARE%: EVAL@COMPILE

[PUTDEF '\M44PAGEBUFFER 'RESOURCES '(NEW (NCREATE 'VMEMPAGEP]
)

(/SETTOPVAL '\M44PAGEBUFFER.GLOBALRESOURCE NIL)

(/SETTOPVAL '\M44PAGEBUFFER.GLOBALRESOURCE NIL)

:: Directory enumeration

(DEFINEQ

(\M44GENERATEFILES

[LAMBDA (FDEV PATTERN DESIREDPROPS OPTIONS) (* bvm%: "12-Oct-85 15:13")

:: Returns a file-generator object that will generate AT LEAST all files in the sys-dir of FDEV whose names match PATTERN. Clients might need to
:: provide additional filtering. For M44, the generate state consists of the HOSTNAME (DSK) followed by a 'search state', a directory pointer and a
:: character list of the sort that \SEARCHDIR1 expects. DIRPTR is the position of the next file to be considered in the directory.

(PROG ((DIRSTREAM (fetch (M44DEVICE SYSDIROFD) of FDEV))
(SORT? (EQMEMB 'SORT OPTIONS))
(CASEBASE (.DISKCASEARRAY.))
(EXT '*))
HOSTNAME NAME VERSION CHARLIST GENSTREAM FILTER DESIREDVERSION SEARCHSTATE HOSTPREFIX)
(OR DIRSTREAM (RETURN (\NULLFILEGENERATOR)))
[COND
((for TAIL on (UNPACKFILENAME.STRING PATTERN) by (CDDR TAIL)
do (SELECTQ (CAR TAIL)
(HOST (SETQ HOSTNAME (CADR TAIL)))
(NAME (SETQ NAME (CADR TAIL)))
(EXTENSION (SETQ EXT (CADR TAIL)))
(VERSION (COND
((OR (EQ [NCHARS (SETQ VERSION (MKATOM (CADR TAIL])
0)
(EQ VERSION 0)) ; Newest version only
(SETQ SORT? T) ; Can only get highest version by sorting
(SETQ VERSION NIL)
(SETQ DESIREDVERSION T))
((SMALLP VERSION) ; An actual specific version to look for
(SETQ DESIREDVERSION VERSION))
(NEQ VERSION '*)) ; Bogus version
(RETURN T))))))
(RETURN T))) ; Bad file name
(RETURN (\NULLFILEGENERATOR)
(SETQ FILTER (DIRECTORY.MATCH.SETUP (CONCAT NAME '%. EXT ";*"))))
(SETQ CHARLIST (for C instring (COND
((OR (EQ 0 (NCHARS EXT))
(EQ (CHCON1 EXT)
(CHARCODE *)))
NAME)
(T (CONCAT NAME '%. EXT))))


```

until (SELCHARQ (SETQ C (\GETBASEBYTE CASEBASE C))
      ((%# *)
      ;; \SEARCHDIR1 currently only checks prefixes, so we truncate at the first * or escape. Also ignore
      ;; version specifications, because of the alternative representations of version 1
      T)
      NIL)
collect C)

(COND
  (DESIREDPROPS ; Create a scratch stream for \M44FILEINFOFN to use
    (SETQ GENSTREAM (create M44STREAM))
    (replace DEVICE of GENSTREAM with FDEV)))
  (SETQ SEARCHSTATE (create M44DIRSEARCHSTATE
    DIRPTR _ 0
    CHARLIST _ CHARLIST))
  (SETQ HOSTPREFIX (CONCAT '{ HOSTNAME '}))
  (RETURN (COND
    (SORT? ; Have to generate the matching files first, sort them, then
            ; enumerate
            (create FILEGENOBJ
              NEXTFILEFN _ (FUNCTION \M44SORTEDNEXTFILEFN)
              FILEINFOFN _ (FUNCTION \M44FILEINFOFN)
              GENFILESTATE _ (create M44GENFILESTATE
                DIROFD _ DIRSTREAM
                SEARCHSTATE _ (\M44SORTFILES DIRSTREAM SEARCHSTATE
                              FILTER DESIREDVERSION HOSTPREFIX
                              (LENGTH CHARLIST))
                GENVERSION _ DESIREDVERSION
                GENSTREAM _ GENSTREAM)))
            T
            (create FILEGENOBJ
              NEXTFILEFN _ (FUNCTION \M44NEXTFILEFN)
              FILEINFOFN _ (FUNCTION \M44FILEINFOFN)
              GENFILESTATE _
                (create M44GENFILESTATE
                  DIROFD _ DIRSTREAM
                  SEARCHSTATE _ SEARCHSTATE
                  GENFILTER _ FILTER
                  GENVERSION _ DESIREDVERSION
                  HOSTNAME _ HOSTPREFIX
                  GENSTREAM _ GENSTREAM))
            ; Order not important
            )))

```

(M44SORTFILES

```

[LAMBDA (DIRSTREAM SEARCHSTATE FILTER DESIREDVERSION HOSTPREFIX PATTERNLENGTH)
  (* bvm%: "7-Jun-84 14:38")
  (SORT (bind FL while (SETQ FL (M44GENERATENEXT DIRSTREAM SEARCHSTATE FILTER DESIREDVERSION HOSTPREFIX
    PATTERNLENGTH))
    collect FL)
  (FUNCTION (LAMBDA (X Y)
    (SELECTQ (UALPHORDER (CAR X)
      (CAR Y))
      (LESSP T)
      (EQUAL (IGREATERP (CADR X)
        (CADR Y))))
    NIL))

```

(M44GENERATENEXT

```

[LAMBDA (DIRSTREAM SEARCHSTATE FILTER DESIREDVERSION HOSTPREFIX PATTERNLENGTH GENFILESTATE)
  ; Edited 21-Jan-91 23:53 by jds

```

;;; Produces the next filename from directory DIRSTREAM satisfying SEARCHSTATE and the more constrained FILTER and DESIREDVERSION, or
;;; returns NIL if no more files. HOSTPREFIX is string to put on front, or NIL for names only. PATTERNLENGTH is the length of the pattern in
;;; SEARCHSTATE. GENFILESTATE is a M44GENFILESTATE whose GENSTREAM and ENTRYSTART want to be set appropriately for
;;; \M44FILEINFOFN; if NIL, then the value is returned for SORTFILES in the form (name version entrystart)

```

(PROG ((PATTERNHASDOT (MEMB (CHARCODE %.)
  (fetch (M44DIRSEARCHSTATE CHARLIST) of SEARCHSTATE)))
  SAWDOT ENTRYSTART TEMP PREFIXLEN TOTALEN THISVERSION RESULT INDEX)
  LP (COND
    ((NOT (SETQ TEMP (M44SEARCHDIR DIRSTREAM SEARCHSTATE)))
      ; Enumeration finished
      (RETURN NIL)))
    (SETQ SAWDOT PATTERNHASDOT)
    (SETQ ENTRYSTART (IDIFFERENCE (GETFILEPTR DIRSTREAM)
      (IPLUS PATTERNLENGTH 13))) ; Read all the characters from the directory
    (SETQ TOTALEN (IPLUS PATTERNLENGTH (SUB1 TEMP))) ; The SUB1 is because the last character is the undesired dot
    (for I from (SUB1 TEMP) to 1 by -1 do
      (SELCHARQ (\BIN DIRSTREAM)
        (! [RETURN (SETQ THISVERSION (M44READVERSION DIRSTREAM
          (SUB1 I))
          (%. (SETQ SAWDOT T))
          NIL)
        finally (SETQ THISVERSION 1))
    (COND
      ((AND DESIREDVERSION (NEQ THISVERSION DESIREDVERSION)

```

```

        (NEQ DESIREDVERSION T)                                ; Failure, try next
        (GO LP)))
[SETQ RESULT (ALLOCSTRING (IPLUS TOTALLEN (SETQ PREFIXLEN (COND
                                                                (HOSTPREFIX (NCHARS HOSTPREFIX))
                                                                (T 0)))
                            (COND
                              ((AND (EQ THISVERSION 1)
                                     HOSTPREFIX)
                               2)
                              (T 0))
                            (COND
                              (SAWDOT 0)
                              (T 1]
(AND HOSTPREFIX (RPLSTRING RESULT 1 HOSTPREFIX))
(\SETFILEPTR DIRSTREAM (IPLUS ENTRYSTART 13))                ; Now read the whole name
(SETQ INDEX PREFIXLEN)
(for I from TOTALLEN to 1 by -1
  do (SELCHARQ (SETQ TEMP (\BIN DIRSTREAM))
              (%. (SETQ SAWDOT T))
              (! (OR SAWDOT (RPLCHARCODE RESULT (add INDEX 1)
                                                (CHARCODE %.)))
                (SETQ SAWDOT T)
                [COND
                  (HOSTPREFIX (RPLCHARCODE RESULT (add INDEX 1)
                                                  (CHARCODE ;))
                    (to (SUB1 I) do (RPLCHARCODE RESULT (add INDEX 1)
                                                         (COND
                                                           (GENFILESTATE (\BIN DIRSTREAM))
                                                           (T
                                                                ;; Make everything a constant version for benefit of SORT. Will replace with real thing later. The
                                                                ;; constant version is chosen in a way that makes 2-digit versions sort in front of 1-digit versions, etc,
                                                                ;; and single-digit versions come out as ;1 to match the ;1 inserted below
                                                                (IDIFFERENCE (CHARCODE 3)
                                                                I]
                                                                (RETURN))
                                                                NIL)
                                                                (RPLCHARCODE RESULT (add INDEX 1)
                                                                TEMP))
(OR SAWDOT (RPLCHARCODE RESULT (add INDEX 1)
                                (CHARCODE %.)))
(COND
  ((AND (EQ THISVERSION 1)
        HOSTPREFIX)
   (RPLSTRING RESULT (ADD1 INDEX)
                      ";1")))
(COND
  ((AND FILTER (NOT (DIRECTORY.MATCH FILTER RESULT)))
   (GO LP)))
(RETURN (COND
        (GENFILESTATE (replace (M44GENFILESTATE ENTRYSTART) of GENFILESTATE with ENTRYSTART)
                      (replace (M44STREAM DIRINFO) of (fetch (M44GENFILESTATE GENSTREAM) of GENFILESTATE)
                              with NIL)
                      RESULT)
        (T (LIST RESULT THISVERSION ENTRYSTART]))

```

(M44NEXTFILEFN

```

[LAMBDA (GENFILESTATE NAMEONLY)                                ; Edited 21-Jan-91 23:53 by jds
;; GENFILESTATE is the state information from the file-generator object created by M44GENERATEFILES. This function returns the next file
;; name as a string. Returns NIL if no files left. It updates GENFILESTATE so that it will get the following satisfactory file on the next call to this
;; function. --- NAMEONLY => returns the filenames without the semi-colon and version number
(PROG ((DIRSTREAM (fetch (M44GENFILESTATE DIROFD) of GENFILESTATE))
      (SEARCHSTATE (fetch (M44GENFILESTATE SEARCHSTATE) of GENFILESTATE))
      (DESIREDVERSION (fetch (M44GENFILESTATE GENVERSION) of GENFILESTATE))
      (FILTER (fetch (M44GENFILESTATE GENFILTER) of GENFILESTATE))
      (HOSTPREFIX (AND (NOT NAMEONLY)
                      (fetch (M44GENFILESTATE HOSTNAME) of GENFILESTATE)))
      PATTERNLENGTH)
(SETQ PATTERNLENGTH (LENGTH (fetch (M44DIRSEARCHSTATE CHARLIST) of SEARCHSTATE)))
(RETURN (M44GENERATENEXT DIRSTREAM SEARCHSTATE FILTER DESIREDVERSION HOSTPREFIX PATTERNLENGTH
                        GENFILESTATE]))

```

(M44SORTEDNEXTFILEFN

```

[LAMBDA (GENFILESTATE NAMEONLY)                                ; Edited 21-Jan-91 23:51 by jds
(LET ((FILES (fetch (M44GENFILESTATE SEARCHSTATE) of GENFILESTATE))
      THISFILE THISNAME V LEN)
(COND
  ((SETQ THISFILE (CAR FILES))                                ; THISFILE = (name version entryStart)
   (SETQ THISNAME (CAR THISFILE))
   (SETQ V (CADR THISFILE))
   ;; need to fill in the correct version number, since the names were generated with constant version number
   (SETQ LEN (NCHARS THISNAME))
  [COND

```

```

[ (ILESSP V 10) ; Easy, 1-digit version
  (\RPLCHARCODE THISNAME LEN (PLUS V (CHARCODE 0))
  (T (SETQ V (CHCON V))
    (for C in V as I from [SETQ LEN (ADD1 (IDIFFERENCE LEN (LENGTH V)
      do (\RPLCHARCODE THISNAME I C]
  (replace (M44STREAM DIRINFO) of (fetch (M44GENFILESTATE GENSTREAM) of GENFILESTATE) with NIL)
  (replace (M44GENFILESTATE ENRYSTART) of GENFILESTATE with (CADDR THISFILE))
  (SETQ FILES (CDR FILES))
  (COND
    ((EQ (fetch (M44GENFILESTATE GENVERSION) of GENFILESTATE)
      T)
      (bind (THISNAMEONLY _ (SUBSTRING THISNAME 1 (SUB1 LEN)))
        while (AND FILES (STRING-EQUAL (SUBSTRING (CAAR FILES)
          1
          (SUB1 LEN))
            THISNAMEONLY))
          do (SETQ FILES (CDR FILES)))
        FILES))
    (replace (M44GENFILESTATE SEARCHSTATE) of GENFILESTATE with FILES)
  THISNAME])

```

(\M44FILEINFOFN

[LAMBDA (GENFILESTATE ATTRIBUTE) ; Edited 21-Jan-91 23:51 by jds

;; Retrieves info of file currently being enumerated. State has a directory pointer to help us out

```

(PROG ((STREAM (fetch (M44GENFILESTATE GENSTREAM) of GENFILESTATE))
  (OR STREAM (RETURN))
  (COND
    ((NULL (fetch (M44STREAM DIRINFO) of STREAM))
      (replace VALIDATION of STREAM with (replace (M44STREAM FILEPAGEMAP) of STREAM with NIL))
      (replace (M44STREAM DIRINFO) of STREAM with (fetch (M44GENFILESTATE ENRYSTART) of GENFILESTATE))
      (replace (M44STREAM FID) of STREAM with (\M44READDIRFID (fetch (M44GENFILESTATE DIROFD) of
        GENFILESTATE
        )
        (fetch (M44GENFILESTATE ENRYSTART) of GENFILESTATE)
        (fetch (M44STREAM FID) of STREAM)))
      (\M44ReadLeaderPage STREAM T)))
    (RETURN (\M44GetFileInfo STREAM ATTRIBUTE]))

```

;; Directory lookup routines

(DEFINEQ

(\M44PARSEFILENAME

[LAMBDA (X RECOG DEV CREATEFLG) ; Edited 21-Jan-91 23:47 by jds

;; This returns a full file specification, with all the information needed to do open, delete, etc. A filespec is a (uname dirptr) pair, with the true
;; version number smashed into the uname. The dirptr is NIL if the file does not currently exist in the directory.

```

(PROG ((DIRSTREAM (fetch (M44DEVICE SYSDIROFD) of DEV))
  UNAME ENDNAMEOFFSET MAYBENEW EXPLICITVERSION FIXEDVERSION UCHARS SOMEPTR TLIST PTR NCHARSLEFT
  BESTVERSION BESTPTR VERS HMIN OLDESTP)
  (COND
    ([NULL (SETQ UNAME (\M44UNPACKFILENAME X DEV (SELECTQ RECOG
      ((NEW OLD/NEW)
        ; We might create a new file here, so tell unpack to save the
        ; original characters.
        (SETQ MAYBENEW T))
      NIL]
      ; BAD FILE NAME

    (RETURN (create FILESPEC
      UNAME _ NIL]

  ;; Name parsed ok, get ready to search directory for it.
  (SETQ UCHARS (fetch (UNAME UCASECHARS) of UNAME))
  (SETQ ENDNAMEOFFSET (+ 13 (LENGTH UCHARS))) ; ENDNAMEOFFSET is length of name we're searching for plus
  ; fixed overhead (header word, FID, name length byte)

  (COND
    (CREATEFLG ; Want to look for a hole, in case we need to create the file. The
    ; 6 is to allow for the maximum number of chars in a version
    ; number

      (SETQ HMIN (FOLDLO (+ ENDNAMEOFFSET 6)
        BYTESPERWORD])

      (SETQ TLIST (CONS 0 UCHARS)) ; Pair of dirptr & chars used to communicate with
      ; \M44SEARCHDIR

      (if (AND (FIXP (SETQ EXPLICITVERSION (fetch (UNAME VERSION) of UNAME)))
        (NEQ EXPLICITVERSION 0))
        then ; If caller gave a real explicit version, then if we find that version,
        ; we know we're done.

          (SETQ FIXEDVERSION EXPLICITVERSION)
          (SETQ OLDESTP (EQ (OR EXPLICITVERSION RECOG)
            'OLDESTP))

        SEARCHLP
        (COND
          ((NULL (SETQ NCHARSLEFT (\M44SEARCHDIR DIRSTREAM TLIST HMIN)))

```

```

; No more prefix matches found
(GO DONE))
(SETQ PTR (\GETFILEPTR DIRSTREAM)) ; Note current position
(COND
  (EQ NCHARSLEFT 1) ; No version, just the final dot remains, so we must have
                    ; matched version 1
  (SETQ VERS 1))
  (NEQ (\BIN DIRSTREAM
        (CHARCODE !)) ; More chars follow before version, so no match
  (GO NEXT))
  ([NULL (SETQ VERS (\M44READVERSION DIRSTREAM (- NCHARSLEFT 2]
  (GO NEXT)))

;; Name matches. VERS is the version number. Is it better than we've seen? Accumulate extreme vers,ptr in BESTVERSION,BESTPTR.
(SETQ PTR (- PTR ENDNAMEOFFSET)) ; Beginning of the directory entry
(COND
  [FIXEDVERSION ; Version must match
   (SETQ BESTPTR PTR) ; Always note a pointer, for benefit of getting case right.
  (COND
    ((EQ VERS FIXEDVERSION) ; The one we've been looking for
     (SETQ BESTVERSION VERS)
     (GO DONE])
    ((OR (NULL BESTVERSION)
         (if OLDESTP
             then (< VERS BESTVERSION)
             else (> VERS BESTVERSION))) ; More extreme than the last one
     (SETQ BESTVERSION VERS)
     (SETQ BESTPTR PTR)))
  NEXT
  (COND
    ((AND HMIN (fetch (M44STREAM DIRHOLEPTR) of DIRSTREAM)) ; Stop looking for a hole if found one
     (SETQ HMIN NIL)))
  (GO SEARCHLP)
  DONE

;; At this point, BESTVERSION is the version, if any, that best matches RECOG or funny version spec in UNAME, i.e., it is the oldest or newest
;; version. BESTPTR is the corresponding directory pointer. In the case where an explicit version was requested but not found, BESTPTR is the
;; directory pointer of SOME version. So now we need to bump the version up for RECOG = NEW, and maybe adjust the characters.
(SETQ SOMEPTR BESTPTR) ; Save dir pointer for getting at true chars for new files.
(if BESTVERSION
  then ; Found one
    (if (if EXPLICITVERSION
            then ; Ignore funny version N when asking for "OLD"
                ; recognition--don't want FOO;N to mean next highest version,
                ; since that's a lie. e.g., it's not infilep.
                (AND (EQ EXPLICITVERSION 'NEW)
                     MAYBENEW)
            else (EQ RECOG 'NEW))
        then (add BESTVERSION 1) ; Bump version, clear directory pointer (since we're creating)
             (SETQ BESTPTR NIL))
    elseif MAYBENEW ; Specified file doesn't exist, but we're willing to create it
      then
        (SETQ BESTVERSION (OR FIXEDVERSION 1))
        (SETQ BESTPTR NIL))
  (RETURN (if BESTVERSION
             then ; Success
               (if (NULL BESTPTR) ; New file. Get the case right if some other version existed.
                   then
                     (if SOMEPTR
                         then (replace (UNAME ORIGCHARS) of UNAME with (\M44READDIRNAME DIRSTREAM
                                                                           SOMEPTR))
                         elseif (fetch (UNAME ORIGCHARS) of UNAME)
                             then ; New recog but existing file--happens when overwriting. Still
                                 ; want to get the characters right.
                                 (replace (UNAME ORIGCHARS) of UNAME with (\M44READDIRNAME DIRSTREAM BESTPTR))
                                 (replace (UNAME VERSION) of UNAME with BESTVERSION)
                                 (create FILESPEC
                                       UNAME _ UNAME
                                       FSDIRPTR _ BESTPTR])

```

(\FINDDIRHOLE

```

[LAMBDA (NWORDS DIRSTREAM) ; Edited 21-Jan-91 23:37 by jds
;; Returns the byte address of a directory hole of size NWORDS. The directory file is positioned just after the 2-byte length field of the hole.
(PROG ((HINT (fetch (M44STREAM DIRHOLEPTR) of DIRSTREAM))
       PTR ENTRYLENGTH C)
  (SETQ PTR (OR HINT 0))
  NEXT
  (\SETFILEPTR DIRSTREAM PTR)
  (COND
    ((\EOF DIRSTREAM)
     (if (AND HINT (> HINT 0))
         then ; Hint failed, so try from the start.
         (SETQ HINT NIL)

```

```

        (SETQ PTR 0)
        (GO NEXT)
    else (GO END))
((AND (>= (SETQ ENTRYLENGTH (+ (LLSH (LOGAND (SETQ C (\BIN DIRSTREAM))
                                          3)
                                  8)
                                (\BIN DIRSTREAM))))
      NWORDS)
 (< C 4))
;; First 6 bits is entry type, next 10 bits are length of entry in words. Free entries have type zero. Thus C < 4 implies this is free entry.
(\SETFILEPTR DIRSTREAM PTR) ; Hole is large enough
[COND
  ((> ENTRYLENGTH NWORDS) ; Too large, so split hole into 2 parts. We'll return the second half
   ; of the hole.
   (\WOUT DIRSTREAM (SETQ ENTRYLENGTH (- ENTRYLENGTH NWORDS)))
   (\SETFILEPTR DIRSTREAM (add PTR (UNFOLD ENTRYLENGTH BYTESPERWORD))
   (GO END)))
  (add PTR (UNFOLD ENTRYLENGTH BYTESPERWORD))
  (GO NEXT)
  END (\WOUT DIRSTREAM NWORDS)
  (RETURN PTR)]

```

(\M44PACKFILENAME

[LAMBDA (UNAME DIRPTR DIRSTREAM) ; Edited 12-Jan-88 12:01 by bvm

;; Produces a Lisp style file-name of the form 'name.[ext];ver'

```

(LET* [(CHARS (OR [AND (NULL *UPPER-CASE-FILE-NAMES*)
                      (OR (fetch (UNAME ORIGCHARS) of UNAME)
                          (if DIRPTR
                              then ; Get the exact name out of the directory
                                (\M44READDIRNAME DIRSTREAM DIRPTR)
                              (fetch (UNAME UCASECHARS) of UNAME)))
          (NAME (CONCAT '{ (fetch (UNAME PARTNAME) of UNAME)
                        ' }
                    (CONCATCODES CHARS)
                    (COND
                      ((MEMB (CHARCODE %.)
                           CHARS)
                       ";")
                      (T ".;"))
                    (fetch (UNAME VERSION) of UNAME)
          (if *UPPER-CASE-FILE-NAMES*
              then (MKATOM NAME)
              else NAME)])

```

(\M44READVERSION

[LAMBDA (DIRSTREAM MAXCHARS) (* bvm%: " 7-Jun-84 11:38")

```

(to MAXCHARS bind (VERSION _ 0)
  C
  do (SETQ C (\BIN DIRSTREAM))
     (COND
      [(AND (IGEQ C (CHARCODE 0))
            (ILEQ C (CHARCODE 9)))]
      (SETQ VERSION (IPLUS (ITIMES VERSION 10)
                          (IDIFFERENCE C (CHARCODE 0)
                          (T ; A non-numeric after a ! means that it wasn't the version marker. This is permissible by alto file spec
                           (RETURN)))
      finally (RETURN VERSION)])

```

(\OPENDISKDESCRIPTOR

[LAMBDA (DEV) ; Edited 21-Jan-91 23:43 by jds

;; Opens and returns a stream on the disk descriptor file for DEV

```

[COND
  ((NOT (type? FDEV DEV))
   (SETQ DEV (\GETDEVICEFROMNAME (fetch (DSKOBJ DISKDEVICENAME) of DEV)
   (OR (fetch (M44DEVICE DDVALID) of DEV)
        (PROG ((OLDD (fetch (M44DEVICE DISKDESCRIPTOROFD) of DEV))
              STREAM)
          (COND
            (OLDD (FORGETPAGES OLDD)))
          [SETQ STREAM (COND
            ((EQ (fetch (M44DEVICE DSKOBJ) of DEV)
                 \MAINDISK)
             (\M44OPENFILEFROMFP DEV "DISKDESCRIPTOR.;1" 'BOTH (\CREATE.FID.FOR.DD DEV)))
            (T (\OPENFILE (CONCAT "{" (fetch (FDEV DEVICENAME) of DEV)
                               "}" "DISKDESCRIPTOR.;1")
                        'BOTH]
            (replace USERVISIBLE of STREAM with NIL)
            (replace (M44DEVICE DISKDESCRIPTOROFD) of DEV with STREAM)
            (replace MAXBUFFERS of STREAM with (ADD1 (fetch EPAGE of STREAM)))
            ; Prepare to buffer the whole file, so that we don't get in trouble
            ; under \NEWPAGE

```

```

      (for I from 0 to (fetch EPAGE of STREAM) do (\MAPPAGE I STREAM))
      ; Ought to define a MAPPAGES to do that more efficiently
      (replace ENDOFSTREAMOP of STREAM with (FUNCTION NIL))
      (replace (M44DEVICE DDVALID) of DEV with T))
(fetch (M44DEVICE DISKDESCRIPTOROFD) of DEV])

```

(\M44READDIRFID

```

[LAMBDA (DIRSTREAM DIRPTR FID) ; Edited 21-Jan-91 23:39 by jds
;; Read the 5-word FID from the directory into FID (or create a new one if FID is nil). Return the new FID.
(\SETFILEPTR DIRSTREAM (+ DIRPTR 2))
(\BINS DIRSTREAM [fetch (FID FIDBLOCK) of (OR FID (SETQ FID (create FID)
0
(UNFOLD 5 BYTESPERWORD))
FID])

```

(\M44READDIRNAME

```

[LAMBDA (DIRSTREAM DIRPTR) ; Edited 11-Jan-88 14:39 by bvm
;; Read the exact file name, sans version number, from directory stream as a list of char codes.
;; Format of a directory entry is --- Type&WordLength (1 word), FP (5 words), Name as a BcplString
(SETFILEPTR DIRSTREAM (+ DIRPTR 12))
(to (SUB1 (BIN DIRSTREAM)) bind CH until (EQ (SETQ CH (BIN DIRSTREAM)
(CHARCODE !))
collect CH])

```

(\M44SEARCHDIR

```

[LAMBDA (STREAM TLIST HMIN) ; Edited 21-Jan-91 23:37 by jds
;; TLIST is a list of the form (POS . NAMECHARS), where POS at entry is a fileptr in the directory file at which to start searching and NAMECHARS
;; is like the characters pairs of a uname. Finds next directory entry for which NAMECHARS is a prefix of the filename. Returns NIL if no entry
;; found, else the length of the remaining chars in the entry. Leaves the directory positioned after the char matching the last char of NAMECHARS
;; --- STREAM is the ofd of the directory file --- At exit, TLIST is smashed so that POS is the fileptr just beyond the found entry. --- if HMIN~NIL,
;; sets STREAM's DIRHOLEPTR to NIL or the fileptr of the first hole of at least HMIN words.
(PROG ((CASEBASE (.DISKCASEARRAY.))
(NEXT (CAR TLIST))
(NAMECHARS (CDR TLIST))
THISNAMELENGTH TARGETLENGTH PTR L TYP ENTRYLENGTH)
(COND
(HMIN (replace (M44STREAM DIRHOLEPTR) of STREAM with NIL)))
(SETQ TARGETLENGTH (LENGTH NAMECHARS))
NEXT
(\SETFILEPTR STREAM (SETQ PTR NEXT))
(COND
((\EOFP STREAM)
(RETURN)))
;; Format of a directory entry is --- Type (0 = hole, 1 = file), 6 bits --- Length of entry in words, 10 bits --- FP 5 words --- Name as a BcplString
(SETQ TYP (\BIN STREAM))
(SETQ ENTRYLENGTH (IPLUS (LLSH (LOGAND TYP 3)
8)
(\BIN STREAM)))
(SETQ NEXT (IPLUS (UNFOLD ENTRYLENGTH BYTESPERWORD)
PTR))
(COND
((NEQ (LRSH TYP 2)
1) ; Not a file
(COND
((AND HMIN (NOT (IGREATERP HMIN ENTRYLENGTH)))
(replace (M44STREAM DIRHOLEPTR) of STREAM with PTR)
(SETQ HMIN NIL)))
(GO NEXT)))
(\SETFILEPTR STREAM (IPLUS PTR 12))
(COND
((ILESSP (SETQ THISNAMELENGTH (\BIN STREAM)
TARGETLENGTH)
(GO NEXT)))
(SETQ L NAMECHARS)
READ
(COND
(NULL L) ; Exhausted the pattern before finding a mismatch, so take it
(RPLACA TLIST NEXT)
(RETURN (IDIFFERENCE THISNAMELENGTH TARGETLENGTH)))
(EQ (\GETBASEBYTE CASEBASE (\BIN STREAM)
(CAR L))
(SETQ L (CDR L))
(GO READ))
(T (GO NEXT])

```

(\M44UNPACKFILENAME

```

[LAMBDA (NAME DEV CREATEFLG) ; Edited 21-Jan-91 23:47 by jds
;; Unpacks file name into a UNAME whose VERSION is the version indicator (either a positive integer or one of OLD, OLDEST, NEW);
;; PARTNAME is the name of DEV. UCASECHARS is a list of uppercase charcodes from the name. If CREATEFLG is true, also sets

```

;; ORIGCHARS to be list of original char codes, for sake of setting real file name

```
(PROG ((CASEBASE (.DISKCASEARRAY.)
      (NC 0)
      J C UPC END ORIGEND VERSION RESULT DOTPREV ORIGDOTPREV EXCESS TAIL)
      (COND
        ((OR (NOT NAME)
              (EQ NAME T)
              (NOT (OR (LITATOM NAME)
                      (STRINGP NAME))))
              (NEQ (NTHCHARCODE NAME 1)
                   (CHARCODE {}))
              (NOT (SETQ J (STRPOS " " NAME 5)))
              (EQ (NTHCHARCODE NAME (add J 1))
                  (CHARCODE <))))
          ;; Name is not a non-null string/atom, or doesn't have a host on front, or { is mismatched, or there's a directory. There used to be
          ;; some junk in here about passing back a different value if the name had a directory than if it was otherwise malformed, but we really
          ;; have no use for that.
          (RETURN NIL)))
      [SETQ END (fetch (UNAME UCASECHARHEAD) of (SETQ RESULT (create UNAME
                                                                    PARTNAME _ (fetch DEVICENAME of DEV]
                                                                    ; End is the cell whose CDR can be smashed.
      (SETQ ORIGEND (fetch (UNAME ORIGCHARHEAD) of RESULT))
      COLLECTNAME
      (COND
        ((NOT (SETQ C (NTHCHARCODE NAME J))) ; End of name
         (GO RET))
        ((EQ (SETQ UPC (\GETBASEBYTE CASEBASE C)
                0) ; Illegal char
         (GO ERR))
        (T [RPLACD END (SETQ END (LIST (SELCHARQ UPC
                                       (GO SEMI))
                                       (%# *) ; Wildcards not allowed
                                       (GO ERR))
                                       (%. ; Omit trailing dots
                                       (PROG1 (SELCHARQ (NTHCHARCODE NAME (ADD1 J))
                                                         (NIL (GO RET))
                                                         ((; !
                                                         (add J 1)
                                                         (GO SEMI))
                                                         UPC)
                                       (SETQ DOTPREV END)
                                       ; Save tail position here in case name gets long
                                       (AND CREATEFLG (SETQ ORIGDOTPREV ORIGEND))))
          UPC]
          [COND
            (CREATEFLG ; Save orig chars as well
             (RPLACD ORIGEND (SETQ ORIGEND (LIST C)
              (add J 1)
              (add NC 1)
              (GO COLLECTNAME)))
          SEMI
```

;; Parsing the stuff after the semicolon; we only accept version, though we do accept the funny symbolic versions H, L and N.

```
(COND
  ([NULL (SETQ C (NTHCHARCODE NAME (add J 1)
    (GO RET))
    ((EQ (SETQ C (\GETBASEBYTE CASEBASE C)
            0) ; Illegal char
    (GO ERR)))
  (SELCHARQ C
    (H (SETQQ VERSION OLD))
    (L (SETQQ VERSION OLDEST))
    (N (SETQQ VERSION NEW))
    (GO COLLECTVERSION))
  (if (EQ J (NCHARS NAME))
    then ; Done
      (GO RET)
    else ; Malformed name
      (GO ERR))
  COLLECTVERSION
  (SETQ VERSION 0)
  [while (AND C (BETWEEN C (CHARCODE 0)
                          (CHARCODE 9)))
    do [SETQ VERSION (+ (TIMES VERSION 10)
                      (- C (CHARCODE 0)
                        (SETQ C (NTHCHARCODE NAME (add J 1)
      (COND
        ((EQ VERSION 0)
         (SETQQ VERSION OLD))
        ((IGREATERP VERSION 65535)
         (GO ERR)))
      (if (NULL C)
        then ; end of name ok
          (GO RET))
      ERR ; BAD FILE NAME
```

```

(RETURN NIL)
RET (replace (UNAME VERSION) of RESULT with VERSION)
[if (> (SETQ EXCESS (- NC (- \MAX.ALTO.NAME.LENGTH 7)))
0)
then
;; Hmm, is name too long? 7 counts for a possible !, 5 version chars and the final dot. This is unnecessarily restrictive for
;; names with shorter versions, but it would get quite untidy if you let version 9 squeak in and then complain or shorten on 10.
;; So best to shorten now. We prefer to leave the extension intact, since that can convey info, and shorten the name.
[if DOTPREV
then (SETQ DOTPREV (CDR DOTPREV)) ; Now (CAR DOTPREV) is the period
(SETQ ORIGDOTPREV (CDR ORIGDOTPREV))
(if (CDR (SETQ TAIL (CL:NTHCDR 10 DOTPREV)))
then ; Extension longer than 10 chars (this allows, e.g.,
; INTERPRESS), so let's shorten it.
(if (<= (SETQ NC (LENGTH (CDR TAIL)))
EXCESS)
then ; Chop off the entire excess
(RPLACD TAIL NIL)
(if CREATEFLG
then (RPLACD (CL:NTHCDR 10 ORIGDOTPREV)
NIL))
(SETQ EXCESS (- EXCESS NC))
else ; only have to get rid of some
(RPLACD (CL:NTHCDR (- NC EXCESS)
TAIL)
NIL)
(if CREATEFLG
then (RPLACD (CL:NTHCDR (+ 10 (- NC EXCESS))
ORIGDOTPREV)
NIL))
(SETQ EXCESS 0]
(if (> EXCESS 0)
then ; Chop away at name
(RPLACD (NLEFT (fetch (UNAME UCASECHARS) of RESULT)
(ADD1 EXCESS)
DOTPREV)
DOTPREV)
(if CREATEFLG
then (RPLACD (NLEFT (fetch (UNAME ORIGCHARS) of RESULT)
(ADD1 EXCESS)
ORIGDOTPREV)
ORIGDOTPREV]
(RETURN RESULT])
)
(RPAQQ \FILENAMECHARSLST (36 43 45 46))
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \FILENAMECHARSLST)
)
(DECLARE%: DONTCOPY
(DECLARE%: EVAL@COMPILE
[RECORD UNAME (VERSION . UCASECHARHEAD)
(RECORD UCASECHARHEAD (ORIGCHARHEAD . UCASECHARS)
(RECORD ORIGCHARHEAD (PARTNAME . ORIGCHARS]
[RECORD FILESPEC (UNAME FSDIRPTR)
(ACCESSFNS FILESPEC ((PNAME (M44PACKFILENAME (fetch UNAME of DATUM]
(RECORD M44GENFILESTATE (DIROFD SEARCHSTATE GENFILTER GENVERSION HOSTNAME GENSTREAM ENTRYSTART))
(RECORD M44DIRSEARCHSTATE (DIRPTR . CHARLIST))
)
(DECLARE%: EVAL@COMPILE
(PUTPROPS BETWEEN MACRO (OPENLAMBDA (V LO HI)
(AND (IGEQ V LO)
(ILEQ V HI))))
)
)
(DEFINEQ
(\CREATE.FID.FOR.DD
[LAMBDA (FDEV) ; Edited 21-Jan-91 23:39 by jds
;; Creates a FID for the file DISKDESCRIPTOR on FDEV, which must be the default disk partition's device
(PROG ((FID (create FID)))
;; Currently \SYSDISK has a copy of the diskdescriptor fp inside it, as looked up by alto at beginning of world, so be lazy and use that
(\BLT (fetch (FID FIDBLOCK) of FID)

```



```
(LOCF (fetch (M44DEVICE DSKDDBLK) of FDEV))
  \LENFP)
(RETURN FID])
```

(\OPENDISK

```
[LAMBDA (PARTNUM FDEV) ; Edited 21-Jan-91 23:32 by jds
  (PROG (DSK DD)
    (OR (\TESTPARTITION PARTNUM)
      (RETURN))
    (SETQ DSK (create DSKOBJ))
    (\LOCKWORDS DSK \NWORDS.DSKOBJ)
    (replace (DSKOBJ DSKPARTITION) of DSK with PARTNUM)
    (replace (DSKOBJ ddPOINTER) of DSK with (LOCF (fetch (DSKOBJ ddLASTSERIAL#) of DSK)))
    (replace (DSKOBJ NDISKS) of DSK with 2)
    (replace (DSKOBJ NTRACKS) of DSK with 406)
    (replace (DSKOBJ NHEADS) of DSK with 2)
    (replace (DSKOBJ NSECTORS) of DSK with 14)
    (replace (DSKOBJ RETRYCOUNT) of DSK with 8)
    (replace (DSKOBJ CBQUEUE) of DSK with (fetch (DSKOBJ CBQUEUE) of \MAINDISK))
    ; Really should have our own
    (RETURN (\OPENDISKDEVICE PARTNUM DSK FDEV])
```

(\OPENDISKDEVICE

```
[LAMBDA (PARTITION DSKOBJ FDEV) ; Edited 21-Jan-91 23:43 by jds
  (DECLARE (GLOBALVARS \MAINDISK) ; Creates the model 44 DSK device and opens its SYSDIR.
  (PROG ([NAME (PACK* 'DSK (OR PARTITION (DISKPARTITION)
    FDEV)
    [OR FDEV (SETQ FDEV (\MAKE.PMAP.DEVICE (create FDEV
      DEVICENAME _ NAME
      NODIRECTORIES _ T
      CLOSEFILE _ (FUNCTION \M44CloseFile)
      DELETEFILE _ (FUNCTION \M44DeleteFile)
      GETFILEINFO _ (FUNCTION \M44GetFileInfo)
      GETFILENAME _ (FUNCTION \M44GetFileName)
      OPENFILE _ (FUNCTION \M44OpenFile)
      READPAGES _ (FUNCTION \M44ReadPages)
      SETFILEINFO _ (FUNCTION \M44SetFileInfo)
      TRUNCATEFILE _ (FUNCTION \M44TruncateFile)
      WRITEPAGES _ (FUNCTION \M44WritePages)
      REOPENFILE _ (FUNCTION \M44OpenFile)
      GENERATEFILES _ (FUNCTION \M44GENERATEFILES)
      EVENTFN _ (FUNCTION \M44EVENTFN)
      DIRECTORYNAMEP _ [FUNCTION (LAMBDA (NAME)
        ; Assume host is OK, check that no directory
        (EQ (NTHCHARCODE NAME -1)
          (CHARCODE ))]
      HOSTNAMEP _ (FUNCTION NIL)
      FREEPAGECOUNT _ (FUNCTION \M44FREEPAGECOUNT)
      OPENP _ (FUNCTION \GENERIC.OPENP)
      REGISTERFILE _ (FUNCTION \ADD-OPEN-STREAM)
      UNREGISTERFILE _ (FUNCTION \GENERIC-UNREGISTER-STREAM]
    (replace (M44DEVICE DSKOBJ) of FDEV with (OR DSKOBJ (SETQ DSKOBJ \MAINDISK)))
    (replace (DSKOBJ DISKDEVICENAME) of DSKOBJ with NAME)
    (RETURN (RESETLST
      (RETSAVE NIL (LIST [FUNCTION (LAMBDA (DEV)
        (COND
          ((NOT (fetch (M44DEVICE DSKPASSWORDOK) of DEV))
            ; Oops, it didn't work, take it away
            (\REMOVEDEVICE DEV]
          (FDEV))
        (\DEFINEDEVICE NAME FDEV) ; have to define it tentatively first so that
        ; \OPENDISKDESCRIPTOR will work
        (COND
          ((\OPENDIR FDEV)
            (COND
              ((NULL PARTITION) ; this is also the default disk
                (\DEFINEDEVICE 'DSK FDEV)))
            FDEV))))))
```

(\OPENDIR

```
[LAMBDA (FDEV) (* bvm%: " 6-APR-83 12:16")
  ;; Opens the model44 directory on the current partition
  (PROG ((PART (fetch (M44DEVICE DSKPARTITION) of FDEV))
    STREAM DD)
    (replace (M44DEVICE DSKPASSWORDOK) of FDEV with NIL)
    (COND
      ((AND (NEQ PART 0)
        (NOT (\TESTPARTITION PART)))
        (replace (M44DEVICE SYSDIROFD) of FDEV with NIL)
        (RETURN)))
      (SETQ STREAM
        (\M44OPENFILEFROMFP FDEV "SYSDIR.;1" 'BOTH
          (create FID
```

W0 _ 32768
W1 _ 100
W2 _ 1
W3 _ 0
W4 _ 1)))

; {DSK}SYSDIR.;1 always has sn 100, leader page on virtual
; page 1

[replace MAXBUFFERS of STREAM with (IMAX 64 (ADD1 (fetch EPAGE of STREAM)

; Enough buffers so that directory is effectively always in core

(replace (M44DEVICE SYSDIROFD) of FDEV with STREAM)
[COND

((NEQ PART 0)
(SETQ DD (\OPENDISKDESCRIPTOR FDEV))
(\SETFILEPTR DD \OFFSET.DISKLASTSERIAL#)
(\BINS DD (LOCF (fetch (M44DEVICE DISKLASTSERIAL#) of FDEV))
0 \NBYTES.DISKINFO)
(add (fetch (M44DEVICE DISKLASTSERIAL#) of FDEV)
3) ; Try to avoid collisions

(COND
((NOT (\M44CHECKPASSWORD FDEV))
(replace (M44DEVICE SYSDIROFD) of FDEV with NIL)
(RETURN)

(replace (M44DEVICE DSKPASSWORDOK) of FDEV with T)
(RETURN STREAM])

(\M44CHECKPASSWORD

[LAMBDA (DEV) (* bvm%: "11-Jun-86 12:20")
(PROG ((STREAM (\OPENFILE (PACK* '{ (fetch (FDEV DEVICENAME) of DEV)
"}SYS.BOOT;1")

'INPUT
'OLD))

PASSVECTOR BUF PASSINFO ASKEDONCE NAME N)
(COND

((NULL STREAM)
(RETURN T)))
(SETQ PASSVECTOR (\ALLOCBLOCK (FOLDHI \NWORDS.BCPLPASSWORD WORDSPERCELL)))
(SETFILEPTR STREAM \OFFSET.BCPLPASSWORD)
(\BINS STREAM PASSVECTOR 0 (UNFOLD \NWORDS.BCPLPASSWORD BYTESPERWORD))
(COND

((EQ (\GETBASE PASSVECTOR 0)
0) ; No password
(\CLOSEFILE STREAM)
(RETURN T)))

(SETFILEPTR STREAM \OFFSET.BCPLUSERNAME)
[SETQ NAME (ALLOCSTRING (SETQ N (\BIN STREAM)
); Read in a bcpl string which is the username installed on the
; disk

(\BINS STREAM (fetch (STRINGP BASE) of NAME)
0 N)
(\CLOSEFILE STREAM)

LP (SETQ PASSINFO (\INTERNAL/GETPASSWORD (fetch (FDEV DEVICENAME) of DEV)
ASKEDONCE NIL NIL NAME))

(COND
(NULL PASSINFO)
(RETURN NIL)))

(COND
((UNINTERRUPTABLY
(SETQ BUF (\GETPACKETBUFFER)) ; HORRIBLE CHEAP TRICK to get some emulator space
(\BLT (\ADDBASE BUF 64)
PASSVECTOR \NWORDS.BCPLPASSWORD)
(SetBcplString (\ADDBASE BUF (IPLUS 64 \NWORDS.BCPLPASSWORD))
(\DECRYPT.PWD (CDR PASSINFO)))
(\CHECKBCPLPASSWORD (\ADDBASE BUF (IPLUS 64 \NWORDS.BCPLPASSWORD))
(\ADDBASE BUF 64)))
(RETURN T))
(T (SETQ ASKEDONCE T)
(GO LP]))

(\M44HOSTNAMEP

[LAMBDA (NAME DEV) (* bvm%: "20-Nov-84 16:06")
(PROG (PARTNUM)
(RETURN (COND

((EQ NAME 'DSK)
(\OPENDISKDEVICE))
((AND (STRPOS 'DSK NAME 1 NIL T)
(SETQ PARTNUM (FIXP (SUBATOM NAME 4)))
(\TESTPARTITION PARTNUM))

(COND
[(EQ PARTNUM (DISKPARTITION))
(RETURN (\GETDEVICEFROMNAME 'DSK]
(T (\OPENDISK PARTNUM])

)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ \OFFSET.BCPLUSERNAME 512)

(RPAQQ \OFFSET.BCPLPASSWORD 768)

(RPAQQ \NWORDS.BCPLPASSWORD 9)

(CONSTANTS \OFFSET.BCPLUSERNAME \OFFSET.BCPLPASSWORD \NWORDS.BCPLPASSWORD)
)
)

:: SYSOUT etc.

(DEFINEQ

(\COPYSYS1

[LAMBDA (STREAM LASTPAGE)

; Edited 21-Aug-88 13:54 by bvm

(COND
((AND (type? M44DEVICE (fetch DEVICE of STREAM))
(EQ (fetch DEVICENAME of (fetch DEVICE of STREAM))
'DSK)
(ERROR "Sysout to Dorado login partition no longer supported.")))
(PROG ((ACTONVMEMFN \VMEMACCESSFN)
(PAGEMAPPED (fetch PAGEMAPPED of (fetch DEVICE of STREAM)))
(NBUFS (SUB1 \#EMUBUFFERS))
(BUFBASE \EMUBUFFERS)
(FIRSTPAGE 2)
(CURSORBAR \EM.CURSORBITMAP)
(CURSORMASK (LLSH 1 (SUB1 BITSPERWORD)))
(DOMINOPAGE (fetch LastDominoFilePage of \InterfacePage))
(DAYBREAKP (EQ \MACHINETYPE \DAYBREAK))
CURSORINC CURSORNEXT NPAGES BUFFERS)

:: Strategy is to copy from the vmem file to STREAM --- The vmem file is read with \ACTONVMEMFILE to finesse the differences among
:: machines. As buffers we use the set of pre-allocated swap buffers, reducing the number available for swapping to a bare minimum of one. If
:: STREAM is pagemapped, we take advantage of knowledge of pagemapped streams to write these buffers directly to the destination stream,
:: which saves the copying that would occur if we just generically used \BOUFS for all streams. In the case of Mod44 DSK, this also lets us use
:: more buffers at a time, because DSK can write directly from the buffers we use for reading the vmem, rather than copying into its own buffers

(RESETSAVE \#SWAPBUFFERS 1) ; Reduce us to one swap buffer, so we can use the rest for
; copying the vmem
(RESETSAVE \EMUSWAPBUFFERS (\ADDBASE BUFBASE (UNFOLD NBUFS WORDSPERPAGE)))
(RESETSAVE \#DISKBUFFERS (COND
((type? M44DEVICE (fetch DEVICE of STREAM))
; DSK code needs 1 extra buffer beyond the ones we give to
; \WRITEPAGES
(SETQ NBUFS (SUB1 NBUFS))
(SETQ BUFBASE (\ADDBASE BUFBASE WORDSPERPAGE))
1)
(T 0)))
(SETQ BUFFERS (to NBUFS as (BUF - BUFBASE) by (\ADDBASE BUF WORDSPERPAGE) collect BUF))
[SETQ CURSORINC (SETQ CURSORNEXT (FOLDLO LASTPAGE (ITIMES 16 16)
; How often to do something to the cursor
(COND
((EQ DOMINOPAGE 0) ; First page to write is the ISF map page, which should be blank
; in a sysout
(\CLEARWORDS BUFBASE WORDSPERPAGE))
(T (CL:FUNCALL ACTONVMEMFN DOMINOPAGE BUFBASE 1)))]
(COND
(PAGEMAPPED (replace EPAGE of STREAM with LASTPAGE) ; Set up end of file correctly. LASTPAGE is last alto page (full),
; which is last Lisp page plus 1
(replace EOFFSET of STREAM with 0)
(\WRITEPAGES STREAM 0 (CAR BUFFERS)))
(T (\BOUFS STREAM (CAR BUFFERS)
0 BYTESPERPAGE)))
(while (<= FIRSTPAGE LASTPAGE)
do [COND
(>= FIRSTPAGE CURSORNEXT) ; Gradually complement the cursor
(\PUTBASE CURSORBAR 0 (LOGXOR (\GETBASE CURSORBAR 0)
CURSORMASK))
(COND
(DAYBREAKP (\DoveDisplay.SetCursorShape)))
(add CURSORNEXT CURSORINC)
(COND
((EQ (SETQ CURSORMASK (LRSH CURSORMASK 1))
0)
(SETQ CURSORBAR (\ADDBASE CURSORBAR 1))
(SETQ CURSORMASK (LLSH 1 (SUB1 BITSPERWORD))
[CL:FUNCALL ACTONVMEMFN FIRSTPAGE BUFBASE (SETQ NPAGES (IMIN NBUFS (ADD1 (- LASTPAGE FIRSTPAGE)
; Read NBUFS pages from vmem, then write them to output
(COND
(NOT PAGEMAPPED) ; Have to just ship the bits
(\BOUFS STREAM BUFBASE 0 (UNFOLD NPAGES BYTESPERPAGE)))
(T (\WRITEPAGES STREAM (SUB1 FIRSTPAGE)
(COND
(< NPAGES NBUFS)
BUFFERS)

```

(T
; Don't write too many pages on the last pass if NPAGES is less
; than length of BUFFERS
(to NPAGES as BUF in BUFFERS collect BUF]
(add FIRSTPAGE NPAGES))
(RETURN NIL])
)

```

:: For MAIKO. \COPYSYS use UNIX-PAGEPERBLOCK.

(DEFINEQ

(MAIKO.CHECKFREESPACE

; Edited 1-Apr-90 18:24 by nm

```

[LAMBDA (FILENAME)
(DECLARE (GLOBALVARS \LDEDESTOVERWRITE \DSKdevice))
(LET ((LASTPAGE (fetch (IFPAGE NActivePages) of \InterfacePage))
(BUFFER (CREATECELL \FIXP))
FULLNAME FILESIZE FREEPAGES HOST)
;; FULLNAME is UNIX/DSK format pathname with UNIX/DSK. And type is string.
[SETQ FULLNAME (if (NULL FILENAME)
then (SETQ HOST 'DSK)
(\UFS.RECOGNIZE.FILE (CONCAT "{" HOST "}" (OR (UNIX-GETENV "LDEDESTSYSOUT")
"/~/.lisp.virtualmem")))
'NON
(\GETDEVICEFROMNAME HOST))
else [SETQ HOST (U-CASE (FILENAMEFIELD FILENAME 'HOST)
(\UFS.RECOGNIZE.FILE FILENAME 'NON (\GETDEVICEFROMNAME HOST)
(SETQ FULLNAME (CONCAT "{" HOST "}" FULLNAME))
;; get current free space
(OR (\UFSGetFreeBlock-C FULLNAME BUFFER)
(LISPERROR "FILE NOT FOUND" FULLNAME))
(if (IGREATERP LASTPAGE (SETQ FREEPAGES (ITIMES BUFFER LISPPAGE-PER-UNIXBLOCK)))
then ;; not enough free space
(if \LDEDESTOVERWRITE
then ;; if possible, try to overwrite
(OR (INFILEP FULLNAME)
(CL:ERROR 'XCL:FS-RESOURCES-EXCEEDED :PATHNAME FULLNAME))
;; file exist, check file size
(SETQ FILESIZE (GETFILEINFO FULLNAME 'SIZE))
(if (IGREATERP LASTPAGE (IPLUS FILESIZE FREEPAGES))
then ;; also, not enough space
(CL:ERROR 'XCL:FS-RESOURCES-EXCEEDED :PATHNAME FULLNAME)
else ;; Remove file, then get enough space to save
(DELFILE FULLNAME))
else (CL:ERROR 'XCL:FS-RESOURCES-EXCEEDED :PATHNAME FULLNAME)])
)
)
(RPAQ? \LDEDESTOVERWRITE NIL)
(DECLARE%: DONTCOPY
(DECLARE%: EVAL@COMPILE
(RPAQQ LISPPAGE-PER-UNIXBLOCK 2)
(CONSTANTS (LISPPAGE-PER-UNIXBLOCK 2))
)
)

```

:: Stats code. On MOD44IO because it writes on the disk and uses records not exported from MOD44IO. (For this and other reasons, GATHERSTATS
:: only works on Dorados.)

(DEFINEQ

(GATHERSTATS

; Edited 21-Jan-91 23:33 by jds

```

[LAMBDA (FILENAME)
;; Enables and disables statistics gathering. Uses low level file operations to avoid stats file being visible from Lisp b/c the file position is not
;; updated as it is written
(DECLARE (GLOBALVARS \STATSON))
(COND
((NEQ \MACHINETYPE \DORADO)
(ERROR "Stats not implemented for this type of machine" FILENAME))
[FILENAME (AND \STATSON (GATHERSTATS))
(SELECTQ (FILENAMEFIELD FILENAME 'HOST)
(DSK)
(NIL (SETQ FILENAME (PACKFILENAME.STRING 'HOST 'DSK 'BODY FILENAME)))
(ERROR "Stats file must be on DSK" FILENAME))
)
)

```

```

      (SETQ \STATSON T)
      (\GATHERSTATS (PROG [(STREAM (\OPENFILE FILENAME 'OUTPUT 'NEW]
                                ; Close before doing stats, cause file isn't really open from Lisp's
                                ; point of view.
                                (RETURN (fetch (ARRAYP BASE) of (fetch (M44STREAM FID)
                                                                    of (PROG1 STREAM
                                                                    (\CLOSEFILE STREAM)
                                                                    (\M44FLUSHDISKDESCRIPTOR (fetch DEVICE
                                                                    of STREAM
                                                                    )))
                                                                    (replace (DSKOBJ DDVALID)
                                                                    of (fetch DEVICE of STREAM)
                                                                    with NIL))]
                                )
      )
      (RPAQQ \STATSON NIL)
      (DECLARE%: EVAL@COMPILE DONTCOPY
      (DECLARE%: DOEVAL@COMPILE DONTCOPY
      (LOCALVARS . T)
      )
      (FILESLOAD (LOADCOMP)
      LLBFS)
      )

```

FUNCTION INDEX

DISKFREEPAGES	14	\M44ExtendFilePageMap ...	4	\M44NEXTFILEFN	18	\M44SORTEDNEXTFILEFN ...	18
GATHERSTATS	28	\M44FILEINFOFN	19	\M44OpenFile	7	\M44SORTFILES	17
\ADDDISKPAGES	12	\M44FillInMap	5	\M44OPENFILEFROMFP	8	\M44TruncateFile	10
\ASSIGNDISKPAGE	13	\M44FLUSHDISKDESCRIPTOR	14	\M44PACKFILENAME	21	\M44UNPACKFILENAME	22
\COPYSYS1	27	\M44FREEPAGECOUNT	15	\M44PARSEFILENAME	19	\M44WriteDiskPage	10
\COUNTDISKFREEPAGES	13	\M44GENERATEFILES	16	\M44READDIRFID	22	\M44WriteLeaderPage	11
\CREATE,FID,FOR,DD	24	\M44GENERATENEXT	17	\M44READDIRNAME	22	\M44WritePages	11
\FINDDIRHOLE	20	\M44GETDATEPROP	6	\M44ReadDiskPage	8	\M44WritePages1	11
\M44AddDiskPages	1	\M44GetFileHandle	5	\M44ReadLeaderPage	8	\MAIKO.CHECKFREESPACE ..	28
\M44CHECKPASSWORD	26	\M44GetFileInfo	5	\M44ReadPages	9	\MAKELEADERDAS	14
\M44CloseFile	2	\M44GetFileName	6	\M44READVERSION	21	\OPENDIR	25
\M44CompleteFH	2	\M44GetPageLoc	6	\M44SEARCHDIR	22	\OPENDISK	25
\M44CREATEFILE	2	\M44HOSTNAMEP	26	\M44SetAccessTimes	9	\OPENDISKDESCRIPTOR ..	21
\M44DeleteFile	3	\M44KillFilePageMap	6	\M44SetEndOfFile	9	\OPENDISKDEVICE	25
\M44DELETEPAGES	12	\M44MAKEDIRENTRY	7	\M44SetFileInfo	9		
\M44EVENTFN	3	\M44MARKPAGEFREE	14	\M44SETFILETYPE	10		

CONSTANT INDEX

LISPPAGE-PER-UNIXBLOCK .	28	\FPROP.TYPE	16	\FPTYPE.UNKNOWN	16	\OFFSET.BCPLPASSWORD ..	27
PageMapIncrement	15	\FPTYPE.BINARY	16	\MAX.ALTO.NAME.LENGTH ..	15	\OFFSET.BCPLUSERNAME ..	27
\FPROP.PAGEMAP	16	\FPTYPE.TEXT	16	\NWORDS.BCPLPASSWORD ..	27		

VARIABLE INDEX

FPROPTYPES	16	\FILENAMECHARSLST	24	\M44MULTFLG	15
FPTYPES	16	\LDEDESTOVERWRITE	28	\STATSON	29

MACRO INDEX

.BFS.TO.LISP.	15	.LISP.TO.BFS.	15	UCASECHAR	15
.DISKCASEARRAY.	15	BETWEEN	24	UPDATEVALIDATION	15

RECORD INDEX

FILESPEC	24	M44DIRSEARCHSTATE	24	M44GENFILESTATE	24
M44DEVICE	15	M44FILEPROP	15	UNAME	24
