

File created: 11-May-2023 21:39:24 {DSK}<cygdrive>c>Users>Larry>home>il>MEDLEY>SOURCES>LEAF.;2

edit by: lmm

changes to: (VARS LEAFCOMPILETIMECOMS)

previous date: 19-Jan-93 10:41:31 {DSK}<cygdrive>c>Users>Larry>home>il>MEDLEY>SOURCES>LEAF.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ LEAFCOMS

(

::: Support for the Leaf random-access filing protocol

(E (RESETSAVE (RADIX 8)))

(COMS :: SEQUIN protocol

(DECLARE%: EVAL@COMPILE DONTCOPY (COMS \* SEQUINCOMS)  
(FILES (LOADCOMP  
TCPHTE))

(INITRECORDS SEQUIN)

(SYSRECORDS SEQUIN)

(FNS CLOSESEQUIN INITSEQUIN GETSEQUIN PUTSEQUIN)

(FNS \SEQUIN.CONTROL \SEQUIN.PUT \SEQUIN.PROCESS \SEQUIN.CLOSE \SEQUIN.FLUSH.CONNECTION  
\SEQUIN.CLEANUP \SEQUIN.FLUSH.RETRANSMIT \SEQUIN.COMPARE \SEQUIN.HANDLE.INPUT  
\SEQUIN.OUT.OF.THE.BLUE \SEQUIN.HANDLE.ACK \SEQUIN.RETRANSMIT \SEQUIN.RETRANSMITNEXT))

(COMS :: LEAF device operations

(FNS \LEAF.CLOSEFILE \LEAF.DELETEFILE \LEAF.DEVICEP \LEAF.RECONNECT \LEAF.DIRECTORYNAMEP  
\LEAF.GENERATEFILES \LEAF.GETFILE \LEAF.PARSE.REMOTE.FILENAME \LEAF.STRIP.QUOTES  
\LEAF.GETFILEDATES \LEAF.GETFILEINFO \LEAF.GETFILEINFO.OPEN \LEAF.GETFILENAME \LEAF.OPENFILE  
\LEAF.READFILENAME \LEAF.ADD.QUOTES \LEAF.READFILEPROP \LEAF.READPAGES \LEAF.REQUESTPAGE  
\LEAF.LOOKUPCACHE CLEAR.LEAF.CACHE LEAF.ASSURE.FINISHED \LEAF.FORCEOUTPUT \LEAF.FLUSH.CACHE  
\LEAF.RENAMEFILE \LEAF.REOPENFILE \LEAF.CREATIONDATE \LEAF.SETCREATIONDATE \LEAF.SETFILEINFO  
\LEAF.SETFILETYPE \LEAF.SETVALIDATION \LEAF.TRUNCATEFILE \LEAF.WRITEPAGES))

(COMS :: Main routing point for LEAF pups

(FNS \SENDLEAF))

(COMS :: Managing LEAF connections

(FNS \OPENLEAFCONNECTION \LEAF.BREAKCONNECTION \CLOSELEAFCONNECTION \LEAF.EVENTFN)  
; This generic fn ought to be on FILEIO

(FNS BREAKCONNECTION))

(COMS :: Functions called when various SEQUIN events occur

(FNS \LEAF.ACKED \LEAF.FIX.BROKEN.SEQUIN \LEAF.REPAIR.BROKEN.PUP \LEAF.USE.NEW.CONNECTION  
\LEAF.RESENDPUPS \LEAF.HANDLE.INPUT \LEAF.OPENERORHANDLER \LEAF.TIMEDIN \LEAF.TIMEDOUT  
\LEAF.NOT.RESPONDING \LEAF.TIMEDOUT.EXCESSIVE \LEAF.ABORT.FROMMENU \LEAF.STREAM.IN.QUEUE  
\LEAF.IDLE \LEAF.MAYBE.FLUSH.CACHE \LEAF.WHENCLOSED \LEAF.IDLE?))

(ADDVARS (NETWORKOSTYPES))

(COMS :: Miscellaneous and error handling

(FNS \ADDLEAFSTRING \FIXPASSWORD \GETLEAFSTRING \IFSERRORSTRING \LEAF.ERROR  
\LEAF.DIRECTORYNAMEONLY GETHOSTINFO GETOSTYPE EXPANDING-PAGEFULLFN)  
(VARS (DEFAULT.OSTYPE 'IFS))  
(GLOBALVARS DEFAULT.OSTYPE))

(COMS :: LookUpFile stuff

(FNS \IFS.LOOKUPFILE)  
(DECLARE%: EVAL@COMPILE DONTCOPY (COMS \* LOOKUPFILECOMS)))

[COMS (FNS \LEAFINIT)

(DECLARE%: DONTVAL@LOAD (P (\LEAFINIT)

(COMS (FNS PRINTLEAF)

(ALISTS (PUPPRINTMACROS 176)))

(INITVARS (LEAFDEBUGFLG)

(LEAFABORTREGION ' (417 616 399 192))

(\MAXLEAFTRIES 4)

(NOFILEPROPEROR)

(DEFAULTFILETYPE 'TEXT)

(\SOCKET.LEAF 35)

(\SEQUIN.TIMEOUTMAX 10000)

(\LEAF.IDLETIMEOUT 1800000)

(\LEAF.CACHETIMEOUT 90000)

(\LEAF.MAXCACHE 10)

(\LEAF.RECOVERY.TIMEOUT 600000)

(\LEAF.MAXLOOKAHEAD 4)

(\FTPAVAILABLE)

(UNIXFTPFLG)

(NONLEAFHOSTS)

(\*UPPER-CASE-FILE-NAMES\* T))

(DECLARE%: EVAL@COMPILE DONTCOPY (COMS \* LEAFCOMPILETIMECOMS))

(INITRECORDS PUPFILESERVER)

(SYSRECORDS PUPFILESERVER))

;;; Support for the Leaf random-access filing protocol
;; SEQUIN protocol

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ SEQUINCOMS
((RECORDS SEQUINPACKET SEQUIN)
(CONSTANTS \* SEQUINOPS)
(CONSTANTS \* SEQUINSTATES)
(CONSTANTS (\SC.EQUAL 0)
(\SC.PREVIOUS 1)
(\SC.DUPLICATE 2)
(\SC.AHEAD 3)
(\SC.OUTOFRANGE 4)
(\PT.SEQUIN 176)
(\SS.NOSOCKET 8)
(\SEQUIN.DEFAULT.ALLOCATION 10)
(\SEQUIN.DEFAULT.RETRANSMITMAX 5))
(MACROS SEQUINOP)))

(DECLARE%: EVAL@COMPILE

[ACCESSFNS SEQUINPACKET ((SEQUINSTART (fetch PUPBASE of DATUM)))
(BLOCKRECORD SEQUINSTART ((NIL 2 WORD)
(ALLOCATE BYTE)
(RECEIVESEQ BYTE)
(SEQCONTROL BYTE)
(SENDSEQ BYTE))
; Pup length, typeword
; Sequin uses ID fields of PUP for control info

]

(DATATYPE SEQUIN ( ;; First: stuff used by SEQUIN level

(SEQNAME POINTER) ; Name of partner
(SEQFRNPORT POINTER) ; Foreign socket
(SEQSOCKET POINTER) ; Local socket
(SEQSTATE BYTE) ; Sequin connection state
(MYSENDSEQ BYTE) ; Number I will next send. These must be byte fields so that they
; will wrap around correctly!
(MYRECEIVESEQ BYTE) ; Number I next expect to receive, i.e. Partner's Send number of
; first unacked packet
(LASTACKEDSEQ BYTE) ; Last Receive seq from partner: all packets with sequence
; numbers before this one have been acked
(SEQOUTALLOC WORD) ; Output allocation: the number of packets I may send without
; their being acked
(SEQINALLOC WORD) ; Input allocation: what I tell my partner
(SEQMAXALLOC WORD) ; The largest I will let output allocation get
(%#UNACKEDSEQS WORD) ; Number of data packets we have sent for which no acks have
; been received
(SEQINPUTLENGTH WORD) ; Number of packets in input (done) queue
(SEQTIMEOUT WORD) ; Timeout before retransmission
(SEQBASETIMEOUT WORD) ; Timeout for this connection in general
(SEQRETRANSMITMAX WORD) ; How many times to retransmit before complaining
(%#SEQRESTARTS WORD) ; Some statistical info...
(%#SEQRETRANSMITS WORD)
(%#SEQDUPLICATES WORD)
(%#SEQTIMEOUTS WORD)
(%#SEQTURNOVERS WORD)
(SEQRETRANSMITQ POINTER) ; Sequin output queue
(SEQTIMER POINTER)
(SEQPROCESS POINTER)
(SEQIGNOREDUPLICATES FLAG)
(SEQRETRANSMITTING FLAG)
(SEQCLOSEME FLAG)
(SEQCLOSEDFORLOGOUT FLAG)
(SEQLASTRESTARTTIMER POINTER) ; Allows for some aging of the connection timeout
(SEQLASTRESTART POINTER)
(SEQRETRANSMITNEXT POINTER)
(SEQEVENT POINTER) ; Signaled when there is input, state changed, or allocation
; changed
; Monitor lock for this structure

;; Second-level functions invoked by SEQUIN

(SEQACKED POINTER) ; (PUP SEQUIN) called when PUP is acked
(SEQINPUT POINTER) ; (PUP SEQUIN) called when PUP arrives as input data
(SEQBROKEN POINTER) ; (SEQUIN PUP) called when a BROKEN sequin arrives (PUP =
; NIL) or attempt to send PUP on broken connection
(SEQABORTED POINTER) ; (SEQUIN) called when PUP arrives with outlandish sequence
; numbers
(SEQTIMEDOUT POINTER) ; (SEQUIN) called when about to retransmit
; SEQRETRANSMITMAX times
(SEQCLOSED POINTER) ; (SEQUIN) called when a connection is flushed, but before its
; retransmit queue is flushed
(SEQIDLETIMEOUTCOMPUTER POINTER) ; Computes timeout before calling SEQIDLEFN when no activity
; on connection. T means forever, NIL means don't

```

        (SEQIDLEFN POINTER)                                ; Called when nothing otherwise is happening, after timeout of
                                                         ; SEQIDLETIMEOUT

        ;; Stuff used by clients of SEQUIN, in particular, LEAF

        (SEQDONEQ POINTER)                                ; Sequins acked but kept around for further handling
        (NIL POINTER)
        (NIL POINTER)
        (LEAFCACHEDFILE POINTER)                          ; Last file accessed, to speed up repeated lookups of same
                                                         ; name
        (LEAFCACHETIMER POINTER)                          ; To timeout the cache
        (LEAFCACHEHITS WORD)
        (LEAFCACHEMISSES WORD)
        (LEAFTIMEOUTCOUNT WORD)
        (LEAFCLOSING FLAG)
        (LEAFOPENCLOSELOCK POINTER)                       ; Monitor lock to keep GETFILE and CLOSEFILE from stepping
                                                         ; on each other

        (LEAFABORTBUTTONWINDOW POINTER)
        (LEAFABORTSTATUS POINTER)
        (LEAFTIMEOUTSTATUS POINTER)
        (SEQTIMEDIN POINTER)
        (NIL POINTER)
        (SEQOPENERORHANDLER POINTER)                     ; (SEQUIN PUP) called on errors trying to open connection

    )
    SEQSTATE _ \SS.UNOPENED SEQOUTALLOC _ 1 SEQINALLOC _ \SEQUIN.DEFAULT.ALLOCATION SEQRETRANSMITMAX _
    \SEQUIN.DEFAULT.RETRANSMITMAX SEQRETRANSMITQ _ (NCREATE 'SYSQUEUE)
    SEQTIMEOUT _ \ETHERTIMEOUT SEQBASETIMEOUT _ \ETHERTIMEOUT SEQTIMER _ (\CREATECELL \FIXP)
    SEQLASTRESTARTTIMER _ (\CREATECELL \FIXP)
    SEQMAXALLOC _ 10 SEQACKED _ (FUNCTION NIL)
    SEQBROKEN _ (FUNCTION NIL)
    SEQABORTED _ (FUNCTION NIL)
    SEQABORTED _ (FUNCTION NIL)
    SEQTIMEDOUT _ (FUNCTION NIL)
    SEQCLOSED _ (FUNCTION NIL)
    SEQIDLETIMEOUTCOMPUTER _ (FUNCTION NIL)
    SEQIDLEFN _ (FUNCTION NIL)
    SEQTIMEDIN _ (FUNCTION NIL)
    SEQOPENERORHANDLER _ (FUNCTION NIL)
    (SYNONYM SEQDONEQ (INPUTQ))
)

(/DECLAREDATATYPE 'SEQUIN
  '(POINTER POINTER POINTER BYTE BYTE BYTE BYTE WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD
    WORD POINTER POINTER POINTER FLAG FLAG FLAG FLAG POINTER POINTER POINTER POINTER POINTER POINTER
    POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
    WORD WORD WORD FLAG POINTER POINTER POINTER POINTER POINTER POINTER POINTER)

  ;; ---field descriptor list elided by lister---
  ' 80)

(RPAQQ SEQUINOPS
  ((\SEQUIN.DATA 0)
   (\SEQUIN.ACK 1)
   (\SEQUIN.NOOP 2)
   (\SEQUIN.RESTART 3)
   (\SEQUIN.OPEN 5)
   (\SEQUIN.BREAK 6)
   (\SEQUIN.OBSOLETE.CLOSE 7)
   (\SEQUIN.DESTROY 9)
   (\SEQUIN.DALLYING 10)
   (\SEQUIN.QUIT 11)
   (\SEQUIN.BROKEN 12)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \SEQUIN.DATA 0)

(RPAQQ \SEQUIN.ACK 1)

(RPAQQ \SEQUIN.NOOP 2)

(RPAQQ \SEQUIN.RESTART 3)

(RPAQQ \SEQUIN.OPEN 5)

(RPAQQ \SEQUIN.BREAK 6)

(RPAQQ \SEQUIN.OBSOLETE.CLOSE 7)

(RPAQQ \SEQUIN.DESTROY 9)

(RPAQQ \SEQUIN.DALLYING 10)

(RPAQQ \SEQUIN.QUIT 11)

(RPAQQ \SEQUIN.BROKEN 12)

```

```

{MEDLEY}<sources>LEAF.;1

(CONSTANTS (\SEQUIN.DATA 0)
  (\SEQUIN.ACK 1)
  (\SEQUIN.NOOP 2)
  (\SEQUIN.RESTART 3)
  (\SEQUIN.OPEN 5)
  (\SEQUIN.BREAK 6)
  (\SEQUIN.OBSOLETE.CLOSE 7)
  (\SEQUIN.DESTROY 9)
  (\SEQUIN.DALLYING 10)
  (\SEQUIN.QUIT 11)
  (\SEQUIN.BROKEN 12))
)

(RPAQQ SEQUINSTATES ((\SS.UNOPENED 0)
  (\SS.OPEN 1)
  (\SS.DALLYING 2)
  (\SS.ABORT 3)
  (\SS.DESTROYED 4)
  (\SS.TIMEDOUT 5)
  (\SS.CLOSING 6)
  (\SS.OPENING 7)
  (\SS.CLOSED 8)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \SS.UNOPENED 0)

(RPAQQ \SS.OPEN 1)

(RPAQQ \SS.DALLYING 2)

(RPAQQ \SS.ABORT 3)

(RPAQQ \SS.DESTROYED 4)

(RPAQQ \SS.TIMEDOUT 5)

(RPAQQ \SS.CLOSING 6)

(RPAQQ \SS.OPENING 7)

(RPAQQ \SS.CLOSED 8)

(CONSTANTS (\SS.UNOPENED 0)
  (\SS.OPEN 1)
  (\SS.DALLYING 2)
  (\SS.ABORT 3)
  (\SS.DESTROYED 4)
  (\SS.TIMEDOUT 5)
  (\SS.CLOSING 6)
  (\SS.OPENING 7)
  (\SS.CLOSED 8))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \SC.EQUAL 0)

(RPAQQ \SC.PREVIOUS 1)

(RPAQQ \SC.DUPLICATE 2)

(RPAQQ \SC.AHEAD 3)

(RPAQQ \SC.OUTOFRANGE 4)

(RPAQQ \PT.SEQUIN 176)

(RPAQQ \SS.NOSOCKET 8)

(RPAQQ \SEQUIN.DEFAULT.ALLOCATION 10)

(RPAQQ \SEQUIN.DEFAULT.RETRANSMITMAX 5)

(CONSTANTS (\SC.EQUAL 0)
  (\SC.PREVIOUS 1)
  (\SC.DUPLICATE 2)
  (\SC.AHEAD 3)
  (\SC.OUTOFRANGE 4)
  (\PT.SEQUIN 176)
  (\SS.NOSOCKET 8)
  (\SEQUIN.DEFAULT.ALLOCATION 10)
  (\SEQUIN.DEFAULT.RETRANSMITMAX 5))
)

(DECLARE%: EVAL@COMPILE

```

```

(PUTPROPS SEQUINOP MACRO ((SEQ OP . ARGS)
                          (APPLY* (fetch (SEQUIN OP) of SEQ) . ARGS)))
)

(FILELOAD (LOADCOMP)
          TCPHTE)
)

(/DECLAREDATATYPE 'SEQUIN
  '(POINTER POINTER POINTER BYTE BYTE BYTE BYTE WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD
    WORD POINTER POINTER POINTER FLAG FLAG FLAG FLAG POINTER POINTER POINTER POINTER POINTER POINTER
    POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
    WORD WORD WORD FLAG POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  '80)

(ADDTOVAR SYSTEMRECLST
  (DATATYPE SEQUIN ((SEQNAME POINTER)
                    (SEQFRNPORT POINTER)
                    (SEQSOCKET POINTER)
                    (SEQSTATE BYTE)
                    (MYSENDSEQ BYTE)
                    (MYRECEIVESEQ BYTE)
                    (LASTACKEDSEQ BYTE)
                    (SEQOUTALLOC WORD)
                    (SEQINALLOC WORD)
                    (SEQMAXALLOC WORD)
                    (%#UNACKEDSEQS WORD)
                    (SEQINPUTQLength WORD)
                    (SEQTIMEOUT WORD)
                    (SEQBASETIMEOUT WORD)
                    (SEQRETRANSMITMAX WORD)
                    (%#SEQRESTARTS WORD)
                    (%#SEQRETRANSMITS WORD)
                    (%#SEQDUPLICATES WORD)
                    (%#SEQTIMEOUTS WORD)
                    (%#SEQTURNOVERS WORD)
                    (SEQRETRANSMITQ POINTER)
                    (SEQTIMER POINTER)
                    (SEQPROCESS POINTER)
                    (SEQIGNOREDUPPLICATES FLAG)
                    (SEQRETRANSMITTING FLAG)
                    (SEQCLOSEME FLAG)
                    (SEQCLOSEDFORLOGOUT FLAG)
                    (SEQLASTRESTARTTIMER POINTER)
                    (SEQLASTRESTART POINTER)
                    (SEQRETRANSMITNEXT POINTER)
                    (SEQEVENT POINTER)
                    (SEQLOCK POINTER)
                    (SEQACKED POINTER)
                    (SEQINPUT POINTER)
                    (SEQBROKEN POINTER)
                    (SEQABORTED POINTER)
                    (SEQTIMEDOUT POINTER)
                    (SEQCLOSED POINTER)
                    (SEQIDLETIMEOUTCOMPUTER POINTER)
                    (SEQIDLEFN POINTER)
                    (SEQDONEQ POINTER)
                    (NIL POINTER)
                    (NIL POINTER)
                    (LEAFCACHEDFILE POINTER)
                    (LEAFCACHETIMER POINTER)
                    (LEAFCACHEHITS WORD)
                    (LEAFCACHEMISSES WORD)
                    (LEAFTIMEOUTCOUNT WORD)
                    (LEAFCLOSING FLAG)
                    (LEAFOPENCLOSELOCK POINTER)
                    (LEAFABORTBUTTONWINDOW POINTER)
                    (LEAFABORTSTATUS POINTER)
                    (LEAFTIMEOUTSTATUS POINTER)
                    (SEQTIMEDIN POINTER)
                    (NIL POINTER)
                    (SEQOPENERRORHANDLER POINTER))))
)

```

(DEFINEQ

**(CLOSESEQUIN**

[LAMBDA (SEQUIN)

; Edited 24-May-91 14:51 by jds

;;; Function called to initiate a close connection for a sequin.

```

(PROG NIL
  (\SEQUIN.CLOSE SEQUIN)
  BLK (AWAIT.EVENT (fetch (SEQUIN SEQEVENT) of SEQUIN)
        \ETHERTIMEOUT)
        (SELECTC (fetch (SEQUIN SEQSTATE) of SEQUIN)

```

```

(\SS.CLOSED (RETURN T))
(\SS.CLOSING NIL)
(RETURN NIL))
(GO BLK])

```

(INITSEQUIN

```

[LAMBDA (SEQUIN PROCNAME) ; Edited 24-May-91 14:51 by jds
  (replace (SEQUIN SEQSOCKET) of SEQUIN with (OPENPUPSOCKET))
  (replace (SEQUIN SEQSTATE) of SEQUIN with \SS.UNOPENED)
  (replace (SEQUIN SEQLOCK) of SEQUIN with (CREATE.MONITORLOCK PROCNAME))
  (replace (SEQUIN SEQEVENT) of SEQUIN with (CREATE.EVENT PROCNAME))
  (replace (SEQUIN MYSENDSEQ) of SEQUIN with 0)
  (replace (SEQUIN MYRECEIVESEQ) of SEQUIN with 0)
  (replace (SEQUIN LASTACKEDSEQ) of SEQUIN with 0)
  (replace (SEQUIN SEQOUTALLOC) of SEQUIN with 1)
  (replace (SEQUIN %UNACKEDSEQS) of SEQUIN with 0)
  (replace (SEQUIN %SEQRESTARTS) of SEQUIN with 0)
  (replace (SEQUIN %SEQDUPLICATES) of SEQUIN with 0)
  (replace (SEQUIN %SEQTIMEOUTS) of SEQUIN with 0)
  (replace (SEQUIN %SEQRETRANSMITS) of SEQUIN with 0)
  (replace (SEQUIN %SEQTURNOVERS) of SEQUIN with 0)
  (replace (SEQUIN SEQPROCESS) of SEQUIN with (ADD.PROCESS (LIST '\SEQUIN.PROCESS SEQUIN)
    'NAME PROCNAME 'RESTARTABLE 'SYSTEM 'AFTEREXIT 'DELETE]))

```

(GETSEQUIN

```

[LAMBDA (SEQUIN) (* bvm%: "10-APR-83 13:26")

```

;;; Function to receive sequin packets on SEQUIN.

```

(PROG (PACKET)
  CL:LOOP
  (COND
    ((SETQ PACKET (\DEQUEUE (fetch (SEQUIN INPUTQ) of SEQUIN)))
      (* (add (fetch (SEQUIN INPUTC) of SEQUIN) -1))
      (* (SEQUIN/CONTROL SEQUIN \SEQUIN.ACK)))
    (RETURN PACKET))
  (EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
    \SS.OPEN)
  (BLOCK)
  (GO CL:LOOP))
(T (RETURN]))

```

(PUTSEQUIN

```

[LAMBDA (SEQUIN OPUP DONTWAIT) ; Edited 24-May-91 14:52 by jds
  (PROG1 (WITH.MONITOR (fetch (SEQUIN SEQLOCK) of SEQUIN)
    (until (AND (SELECTC (fetch (SEQUIN SEQSTATE) of SEQUIN)
      (\SS.OPEN (replace (SEQUINPACKET SEQCONTROL) of OPUP with \SEQUIN.DATA)
        T)
      (\SS.UNOPENED (replace (SEQUIN SEQSTATE) of SEQUIN with \SS.OPENING)
        (replace (SEQUINPACKET SEQCONTROL) of OPUP with \SEQUIN.OPEN)
        T)
      (\SS.OPENING NIL)
      (RETURN (PUTSEQUIN (OR (SEQUINOP SEQUIN SEQBROKEN SEQUIN OPUP)
        (RETURN OPUP))
        OPUP)))
    (ILESSP (fetch (SEQUIN %UNACKEDSEQS) of SEQUIN)
      (fetch (SEQUIN SEQOUTALLOC) of SEQUIN))
    (ILEQ (fetch (SEQUIN SEQINPUTQLength) of SEQUIN)
      (fetch (SEQUIN SEQINALLOC) of SEQUIN))
    (COND
      ((NOT (fetch (SEQUIN SEQRETRANSMITTING) of SEQUIN))
        T)
      (T ;; Should never happen, because \SEQUIN.PROCESS does not relinquish the lock. Test is here for
        ;; debugging
        (COND
          (LEAFDEBUGFLG (HELP "lock obtained while retransmitting" SEQUIN)))
        NIL)))
    do (COND
      (DONTWAIT (RETURN)))
      (MONITOR.AWAIT.EVENT (fetch (SEQUIN SEQLOCK) of SEQUIN)
        (fetch (SEQUIN SEQEVENT) of SEQUIN)
        \ETHERTIMEOUT)
    finally (\SEQUIN.PUT SEQUIN OPUP T)
      (RETURN SEQUIN)))
  (BLOCK])
)

```

(DEFINEQ

(\SEQUIN.CONTROL

```

[LAMBDA (SEQUIN CONTROL PUP) ; Edited 23-Dec-87 16:42 by bvm:

```

;;; Routine to send a control sequin of type CONTROL to the other end

```
[COND
  (PUP
    (\CLEARBYTES (LOCF (fetch PUPSOURCE of PUP)
      0 6))
    (T (SETQ PUP (ALLOCATE.PUP)
      (replace PUPLENGTH of PUP with \PUPOVLEN)
      (replace (SEQUINPACKET SEQCONTROL) of PUP with CONTROL)
      (\SEQUIN.PUT SEQUIN PUP]))
    ; Clear source net,host,socket so that SENDPUP will fill them in
    ; with the truth.
```

(\SEQUIN.PUT

```
[LAMBDA (SEQUIN PUP ISDATA) ; Edited 24-May-91 14:52 by jds
  (replace PUPTYPE of PUP with \PT.SEQUIN)
  (replace PUPDEST of PUP with (CAR (fetch (SEQUIN SEQFRNPORT) of SEQUIN)))
  (replace PUPDESTSOCKET of PUP with (CDR (fetch (SEQUIN SEQFRNPORT) of SEQUIN)))
  (UNINTERRUPTABLY
    (PROG ((SENDSEQ (fetch (SEQUIN MYSENDSEQ) of SEQUIN))
      (replace (SEQUINPACKET RECEIVESEQ) of PUP with (fetch (SEQUIN MYRECEIVESEQ) of SEQUIN))
      (replace (SEQUINPACKET SENDSEQ) of PUP with SENDSEQ)
      [COND
        (ISDATA [replace (SEQUIN MYSENDSEQ) of SEQUIN with (COND
          ((EQ SENDSEQ 255)
            (add (fetch (SEQUIN %#SEQTURNOVERS) of SEQUIN)
              1)
            0)
          (T (ADD1 SENDSEQ)
            ; Data packets increment the send sequence, and we have to keep them around for possible retransmission
            (replace EPREQUEUE of PUP with (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN))
            (add (fetch (SEQUIN %#UNACKEDSEQS) of SEQUIN)
              1))
          (T (replace EPREQUEUE of PUP with 'FREE]
            (replace (SEQUINPACKET ALLOCATE) of PUP with (fetch (SEQUIN SEQINALLOC) of SEQUIN))
            (SENDPUP (fetch (SEQUIN SEQSOCKET) of SEQUIN)
              PUP)
            (\CLOCK0 (fetch (SEQUIN SEQTIMER) of SEQUIN))
            ; Make sure the SEQUIN watcher runs. It might be in its long idle phase, and if no packets arrive on its socket, it won't wake up to notice
            ; that remote host is not responding
            (WAKE.PROCESS (fetch (SEQUIN SEQPROCESS) of SEQUIN))))]))])
```

(\SEQUIN.PROCESS

```
[LAMBDA (SEQUIN) ; Edited 24-May-91 14:52 by jds
  (DECLARE (SPECVARS SEQUIN))
  (WITH.MONITOR (fetch (SEQUIN SEQLOCK) of SEQUIN)
    (RESETSAVE NIL (LIST (FUNCTION \SEQUIN.CLEANUP)
      SEQUIN))
    [PROCESSPROP (THIS.PROCESS)
      'INFOHOOK
      (FUNCTION (LAMBDA NIL
        (INSPECT SEQUIN]
      (PROG ((SOC (fetch (SEQUIN SEQSOCKET) of SEQUIN))
        (RETRANSQ (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN))
        (CNT 0)
        RETRANSMITINCREMENT PUP SOCEVENT TIMEOUT REASON)
        (COND
          ((NOT SOC) ; Sequin was killed
            (RETURN)))
        (SETQ SOCEVENT (PUPSOCKETEVENT SOC))
        LP [COND
          ((fetch (SEQUIN SEQCLOSEME) of SEQUIN)
            (RETURN))
          ((SETQ PUP (GETPUP SOC))
            (SELECTC (fetch PUPTYPE of PUP)
              (\PT.SEQUIN (COND
                ((\SEQUIN.HANDLE.INPUT SEQUIN PUP)
                  ; Something interesting happened
                )))
            (\PT.ERROR [COND
              ((EQ PUPTRACEFLG 'PEEK)
                (PRINTPUP PUP 'GET]
              [COND
                ((NEQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
                  \SS.OPENING)
                  (SELECTC (fetch ERRORPUPCODE of PUP)
                    (\PUPE.NOSOCKET ; Connection was open and went away?
                      (SEQUINOP SEQUIN SEQBROKEN SEQUIN))
                    NIL))
                ((SETQ REASON (SEQUINOP SEQUIN SEQOPENERERRORHANDLER SEQUIN PUP))
                  (RELEASE.PUP PUP)
                  (RETURN (\SEQUIN.FLUSH.CONNECTION SEQUIN \SS.ABORT REASON]
```

```

(RELEASE.PUP PUP))
(RELEASE.PUP PUP))
((fetch (SEQUIN SEQRETRANSMITTING) of SEQUIN)
(\SEQUIN.RETRANSMITNEXT SEQUIN))
((EQ (MONITOR.AWAIT.EVENT (fetch (SEQUIN SEQLOCK) of SEQUIN)
SOCEVENT
(OR (SETQ TIMEOUT (AND (EQ (fetch (SEQUIN %UNACKEDSEQS) of SEQUIN)
0)
(NEQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
\SS.CLOSING)
(SEQUINOP SEQUIN SEQIDLETIMEOUTCOMPUTER SEQUIN)))
(fetch (SEQUIN SEQTIMEOUT) of SEQUIN)))
PSTAT.TIMEDOUT) ; Nothing urgent happening
(COND
(TIMEOUT (SEQUINOP SEQUIN SEQIDLEFN SEQUIN))
(T ; Waiting for acks
(COND
((\CLOCKGREATERP (fetch (SEQUIN SEQTIMER) of SEQUIN)
(fetch (SEQUIN SEQTIMEOUT) of SEQUIN))
; Haven't seen anything in a while, so prod the other end
(INCLEAFSTAT (fetch (SEQUIN %SEQTIMEOUTS) of SEQUIN))
[COND
((NEQ (fetch (SEQUIN MYRECEIVESEQ) of SEQUIN)
(fetch (SEQUIN SEQLASTRESTART) of SEQUIN))
; This is the first time we've had trouble at this sequence
(SETQ CNT 1)
(SETQ RETRANSMITINCREMENT (IMAX 2000 (LRSH (fetch (SEQUIN SEQTIMEOUT)
of SEQUIN)
1)))
(replace (SEQUIN SEQLASTRESTART) of SEQUIN with (fetch (SEQUIN MYRECEIVESEQ)
of SEQUIN))
(SETUPTIMER 0 (fetch (SEQUIN SEQLASTRESTARTTIMER) of SEQUIN))
(T (SEQUINOP SEQUIN SEQTIMEDOUT SEQUIN (add CNT 1))
(COND
((fetch (SEQUIN SEQCLOSEME) of SEQUIN)
; In case SEQTIMEDOUT closed the connection
(RETURN]
(COND
((ILESSP (fetch (SEQUIN SEQTIMEOUT) of SEQUIN)
\SEQUIN.TIMEOUTMAX)
(add (fetch (SEQUIN SEQTIMEOUT) of SEQUIN)
RETRANSMITINCREMENT)))
(COND
((EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
\SS.CLOSING)
(\SEQUIN.CONTROL SEQUIN \SEQUIN.DESTROY))
((EQ (fetch (SEQUIN %UNACKEDSEQS) of SEQUIN)
1) ; Only one thing in queue, just resend it
(\SEQUIN.RETRANSMIT SEQUIN))
(T ; All our stuff is acked, but client is still waiting for something; or
; more than one thing
(\SEQUIN.CONTROL SEQUIN \SEQUIN.NOOP]
(BLOCK)
(GO LP))))))

```

```

(\SEQUIN.CLOSE
[LAMBDA (SEQUIN) ; Edited 24-May-91 14:52 by jds
(WITH.MONITOR (fetch (SEQUIN SEQLOCK) of SEQUIN)
(COND
((EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
\SS.OPEN)
(replace (SEQUIN SEQSTATE) of SEQUIN with \SS.CLOSING)
(\SEQUIN.CONTROL SEQUIN \SEQUIN.DESTROY)
T))))))

```

```

(\SEQUIN.FLUSH.CONNECTION
[LAMBDA (SEQUIN FINALSTATE REASON) ; Edited 24-May-91 14:52 by jds

```

;;; Close a sequin connection

```

(PROG ((PROC (fetch (SEQUIN SEQPROCESS) of SEQUIN)))
(COND
((NULL PROC) ; Cleanup has already been done
(RETURN)))
(\SEQUIN.FLUSH.RETRANSMIT SEQUIN)
(replace (SEQUIN SEQSTATE) of SEQUIN with (OR FINALSTATE \SS.ABORT))
(NOTIFY.EVENT (fetch (SEQUIN SEQEVENT) of SEQUIN))
(CLOSEPUPSOCKET (fetch (SEQUIN SEQSOCKET) of SEQUIN))
(replace (SEQUIN SEQSOCKET) of SEQUIN with NIL)
(replace (SEQUIN SEQPROCESS) of SEQUIN with NIL)
(SEQUINOP SEQUIN SEQCLOSED SEQUIN FINALSTATE REASON)
(COND
((NEQ PROC (THIS.PROCESS))
(DEL.PROCESS PROC))
(T (replace (SEQUIN SEQCLOSEME) of SEQUIN with T]))

```



(\SEQUIN.CLEANUP

```
[LAMBDA (SEQUIN) ; Edited 24-May-91 14:52 by jds
  ;; Called via RESETSAVE by Sequin process to perform cleanup if the sequin watcher is killed unexpectedly. Important thing is that we not do this
  ;; on HARDRESET
  (SELECTQ RESETSTATE
    ( (ERROR RESET)
      (COND
        ((EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
              \SS.OPEN)
          (\SEQUIN.CONTROL SEQUIN \SEQUIN.BROKEN)))
        (\SEQUIN.FLUSH.CONNECTION SEQUIN \SS.ABORT))
      NIL])
```

(\SEQUIN.FLUSH.RETRANSMIT

```
[LAMBDA (SEQUIN) ; Edited 24-May-91 14:52 by jds
  (PROG ((REPUP (fetch (SEQUIN SEQRETRANSMITNEXT) of SEQUIN)))
    (COND
      (REPUP (replace (SEQUIN SEQRETRANSMITNEXT) of SEQUIN with NIL)
              (while REPUP do (\ENQUEUE (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN)
                                          (PROG1 REPUP
                                            (SETQ REPUP (fetch EPLINK of REPUP))))))
```

(\SEQUIN.COMPARE

```
[LAMBDA (X Y) (* bvm%: " 6-Jan-85 00:14")
```

;;; Function to return sequence comparison on received pups

```
(PROG ((DIF (LOGAND (IDIFFERENCE X Y)
                    255)))
  (RETURN (COND
    ((EQ DIF 0) \SC.EQUAL)
    ((EQ DIF 255) \SC.PREVIOUS)
    ((IGEQ DIF 192) \SC.DUPLICATE)
    ((ILEQ DIF 64) \SC.AHEAD)
    (T \SC.OUTOFRANGE]))
```

(\SEQUIN.HANDLE.INPUT

```
[LAMBDA (SEQUIN PUP) ; Edited 24-May-91 14:52 by jds
```

;;; Function to handle input pup. Checks that sequence numbers are sensible, takes appropriate action if retransmission needed or releases packets  
;;; that are hereby acked. Hands new data packets off to next-level protocol

```
(PROG (ALLOC NEWACKSEQ)
  (COND
    ((NEQ (fetch (PUP PUPTYPE) of PUP)
          \PT.SEQUIN)
      (RELEASE.PUP PUP)
      (RETURN))
    ((EQ (fetch (SEQUINPACKET SEQCONTROL) of PUP)
         \SEQUIN.BROKEN)
      (SEQUINOP SEQUIN SEQBROKEN SEQUIN)
      (RELEASE.PUP PUP)
      (RETURN)))
    (SELECTC (\SEQUIN.COMPARE (fetch (SEQUINPACKET SENDSEQ) of PUP)
              (fetch (SEQUIN MYRECEIVESEQ) of SEQUIN))
      (\SC.OUTOFRANGE
        (RETURN (\SEQUIN.OUT.OF.THE.BLUE SEQUIN PUP)))
      (\SC.AHEAD
        ; Partner got ahead, ask for retransmission from
        ; MYRECEIVESEQ
        (COND
          ((NEQ (fetch (SEQUINPACKET SEQCONTROL) of PUP)
                \SEQUIN.RESTART)
            ; Don't get into a RESTART loop! Do the retransmit requested by partner and hope that things get better
            (\SEQUIN.CONTROL SEQUIN \SEQUIN.RESTART)
            (RELEASE.PUP PUP)
            (RETURN)))
          (\SC.DUPLICATE
            ; Nothing new, drop it
            (GO DUPLICATE))
          (\SC.PREVIOUS
            ; Retransmission of last packet is simple way to get restart
            (COND
              ((NOT (fetch (SEQUIN SEQIGNOREDUPLICATES) of SEQUIN))
                (replace (SEQUINPACKET SEQCONTROL) of PUP with \SEQUIN.RESTART))
              ((EQ (fetch (SEQUINPACKET SEQCONTROL) of PUP)
                   \SEQUIN.DALLYING)
                ;; KLUDGE!!! To work around bug in Twenex Leaf server. Remove this when server is fixed for enough
                ;; people
```

```

NIL)
      (T (GO DUPLICATE)))
NIL)
[COND
  [(EQ (SETQ ALLOC (fetch (SEQUINPACKET ALLOCATE) of PUP))
    0)
    (COND
      ((ILESSP (fetch (SEQUIN SEQINPUTQLength) of SEQUIN)
        1)
        ;; Allocation = 0 normally defaults to 1; however, in rare cases, my partner has actually decremented its allocation below 1,
        ;; meaning I can't send ANY packets.
        (SETQ ALLOC 1]
      ((IGREATERP ALLOC (fetch (SEQUIN SEQMAXALLOC) of SEQUIN))
        (SETQ ALLOC (fetch (SEQUIN SEQMAXALLOC) of SEQUIN)
[COND
  ((NEQ (fetch (SEQUIN SEQOUTALLOC) of SEQUIN)
    ALLOC)
    (replace (SEQUIN SEQOUTALLOC) of SEQUIN with ALLOC) ; Our allocation changed, maybe someone is waiting to send
    (NOTIFY.EVENT (fetch (SEQUIN SEQEVENT) of SEQUIN)
[SELECTC (\SEQUIN.COMPARE (SETQ NEWACKSEQ (fetch (SEQUINPACKET RECEIVESEQ) of PUP))
  (fetch (SEQUIN LASTACKEDSEQ) of SEQUIN))
  (\SC.OUTOFRANGE
    (RETURN (\SEQUIN.OUT.OF.THE.BLUE SEQUIN PUP)))
  ((LIST \SC.DUPLICATE \SC.PREVIOUS)
    (GO DUPLICATE))
  (\SC.AHEAD ; Release packets acked by this pup
    (\SEQUIN.HANDLE.ACK SEQUIN NEWACKSEQ))
NIL)
(SELECTC (fetch (SEQUINPACKET SEQCONTROL) of PUP)
  (\SEQUIN.DATA (UNINTERRUPTABLY
    (COND
      ((EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
        \SS.OPENING)
        (replace (SEQUIN SEQSTATE) of SEQUIN with \SS.OPEN)))
      (add (fetch (SEQUIN MYRECEIVESEQ) of SEQUIN)
        1)
        (SEQUINOP SEQUIN SEQINPUT PUP SEQUIN)
        (NOTIFY.EVENT (fetch (SEQUIN SEQEVENT) of SEQUIN)))
      (COND
        ((NEQ (fetch (SEQUIN SEQTIMEOUT) of SEQUIN)
          (fetch (SEQUIN SEQBASETIMEOUT) of SEQUIN))
          (replace (SEQUIN SEQTIMEOUT) of SEQUIN with (fetch (SEQUIN SEQBASETIMEOUT)
            of SEQUIN))
            (SEQUINOP SEQUIN SEQTIMEDIN SEQUIN)))
          ; Set timeout back to normal now that we have a response
          (RETURN T))
        (\SEQUIN.RESTART
          (INCLEAFSTAT (fetch (SEQUIN %SEQRESTARTS) of SEQUIN))
          (\SEQUIN.RETRANSMIT SEQUIN))
        (\SEQUIN.DALLYING ; Only sequin Users get this
          (COND
            ((EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
              \SS.CLOSING)
              (\SEQUIN.CONTROL SEQUIN \SEQUIN.QUIT)
              (\SEQUIN.FLUSH.CONNECTION SEQUIN \SS.CLOSED))))
            (\SEQUIN.DESTROY ; Only sequin Servers get this or QUIT
              (\SEQUIN.CONTROL SEQUIN \SEQUIN.DALLYING)
              (replace (SEQUIN SEQSTATE) of SEQUIN with \SS.DALLYING))
            (\SEQUIN.QUIT (COND
              ((EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
                \SS.DALLYING)
                (\SEQUIN.FLUSH.CONNECTION SEQUIN \SS.CLOSED))))
              NIL)
            (RELEASE.PUP PUP)
            (RETURN T)
          DUPLICATE
            (INCLEAFSTAT (fetch (SEQUIN %SEQDUPLICATES) of SEQUIN))
            (RELEASE.PUP PUP)
            (RETURN]))

```

(\SEQUIN.OUT.OF.THE.BLUE [LAMBDA (SEQUIN PUP) (\* bvm%: "27-JUL-83 22:29")

::: Called when PUP arrives on SEQUIN with outlandish sequence numbers

```

(** (replace (SEQUIN SEQSTATE) of SEQUIN with \SS.ABORT)
  (\SEQUIN.CONTROL SEQUIN \SEQUIN.BROKEN) (SEQUINOP SEQUIN SEQABORTED SEQUIN)
  (RELEASE.PUP PUP))

```

NIL])

(\SEQUIN.HANDLE.ACK [LAMBDA (SEQUIN ACKSEQ) ; Edited 24-May-91 14:52 by jds

;;; Function to dispose of Pups on the output queue which have been acknowledged by a Receive sequence of ACKSEQ

```
(bind (QUEUE _ (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN))
      NEWACKSEQ PUP do
  (COND
   ((NULL (SETQ PUP (\QUEUEHEAD QUEUE)))
    ; All packets up to ACKSEQ-1 are now acknowledged
    ; Pup hasn't come back from transmission yet; wait
    (COND
     ((fetch (SEQUIN SEQRETRANSMITTING) of SEQUIN)
      ; Pup hasn't come back yet because we haven't sent it! Send
      ; another
      (\SEQUIN.RETRANSMITNEXT SEQUIN)))
     (BLOCK))
    (UNINTERRUPTABLY
     (\DEQUEUE QUEUE)
     (add (fetch (SEQUIN %UNACKEDSEQS) of SEQUIN)
          -1)
     (replace (SEQUIN LASTACKEDSEQ) of SEQUIN
              with (SETQ NEWACKSEQ (LOGAND (ADD1 (fetch (SEQUINPACKET SENDSEQ)
                                                         of PUP))
                                           255)))
     (SEQUINOP SEQUIN SEQACKED PUP SEQUIN)
     (EQ NEWACKSEQ ACKSEQ))
    (RETURN]))
```

(\SEQUIN.RETRANSMIT

[LAMBDA (SEQUIN)

; Edited 24-May-91 14:52 by jds

;;; Routine to retransmit output sequins

```
(OR (fetch (SEQUIN SEQRETRANSMITTING) of SEQUIN)
    (PROG ((QUEUE (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN)))
          (COND
           ((NULL (fetch SYSQUEUEHEAD of QUEUE))
            (RETURN T)))
          (while (NEQ (LOGAND (ADD1 (fetch (SEQUINPACKET SENDSEQ) of (fetch SYSQUEUEUTAIL of QUEUE)))
                             255)
                    (fetch (SEQUIN MYSENDSEQ) of SEQUIN))
            do ;; Not all of our packets have been transmitted yet; don't restart now or our retransmit queue will get out of order
              (BLOCK))
            (UNINTERRUPTABLY
             (replace (SEQUIN SEQRETRANSMITNEXT) of SEQUIN with (fetch SYSQUEUEHEAD of QUEUE))
             (replace SYSQUEUEHEAD of QUEUE with (replace SYSQUEUEUTAIL of QUEUE with NIL))
             ; Detach chain of pups from retransmit queue so that they can
             ; return there normally
             (replace (SEQUIN SEQRETRANSMITTING) of SEQUIN with T))))
```

(\SEQUIN.RETRANSMITNEXT

[LAMBDA (SEQUIN)

; Edited 24-May-91 14:52 by jds

```
(PROG ((NEXTPOP (fetch (SEQUIN SEQRETRANSMITNEXT) of SEQUIN)))
      (replace EPREQUEUE of NEXTPOP with (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN))
      (replace (SEQUINPACKET RECEIVESEQ) of NEXTPOP with (fetch (SEQUIN MYRECEIVESEQ) of SEQUIN))
      (replace (SEQUINPACKET ALLOCATE) of NEXTPOP with (fetch (SEQUIN SEQINALLOC) of SEQUIN))
      [SENDPOP (fetch (SEQUIN SEQSOCKET) of SEQUIN)
              (PROG1 NEXTPOP
                 (OR (replace (SEQUIN SEQRETRANSMITNEXT) of SEQUIN with (fetch EPLINK of NEXTPOP))
                     (replace (SEQUIN SEQRETRANSMITTING) of SEQUIN with NIL)))]
      (add (fetch (SEQUIN %SEQRETRANSMITS) of SEQUIN)
          1))
```

;;; LEAF device operations

(DEFINEQ

(\LEAF.CLOSEFILE

[LAMBDA (STREAM CONNECTION LEAFHANDLE FORCE)

; Edited 2-Nov-92 03:35 by sybalsky:mv:envos

;;; Closes the file open on this LEAF connection. CONNECTION and LEAFHANDLE are obtained from STREAM if necessary; else STREAM may be NIL

```
(PROG (OPUP DATA (INTERNAL CONNECTION))
      [COND
       (STREAM (\CLEARMAP STREAM)
        (OR (SETQ CONNECTION (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM))
            (LISPERROR "FILE NOT OPEN" STREAM))
        (COND
         ((WITH.MONITOR (fetch (SEQUIN LEAFOPENCLOSELOCK) of CONNECTION)
                        [COND
                         ((EQ (fetch (SEQUIN SEQSTATE) of CONNECTION)
                              \SS.OPEN)
                          (COND
```

```

[ (AND (NOT FORCE)
      (NOT (DIRTYABLE STREAM)))
  ; Don't really close it; keep it around in case someone wants to
  ; look at it again soon
  (OR INTERNAL (replace (LEAFSTREAM LEAFREALLYOPEN) of STREAM with NIL))
  ;; If this is a call from CLOSEF then mark the stream as 'really' closed, so that we know we can close it
  ;; later
  (LET ((CACHE (fetch (SEQUIN LEAFCACHEDFILE) of CONNECTION)))
    (COND
      ((NULL CACHE) ; No cache before, so just make this the cached file
       (replace (SEQUIN LEAFCACHEDFILE) of CONNECTION with STREAM)
       T)
      ((EQ CACHE STREAM) ; Closing the already cached file? Do nothing
       T)
      ((EQ (fetch (STREAM FULLFILENAME) of STREAM)
            (fetch (STREAM FULLFILENAME) of CACHE))
       ;; Two streams open on the same file. Could happen if STREAM was opened with an incomplete
       ;; filename. Always prefer to keep the originally cached file around, so fall thru now and close
       ;; STREAM
       NIL)
      (T (replace (SEQUIN LEAFCACHEDFILE) of CONNECTION with STREAM)
        (COND
          ((fetch (LEAFSTREAM LEAFREALLYOPEN) of CACHE)
           T)
          (T ; Close the formerly cached stream if Lisp thinks it is closed
            (SETQ STREAM CACHE)
            NIL]
        ((EQ STREAM (fetch (SEQUIN LEAFCACHEDFILE) of CONNECTION))
         ; We are about to close the cached stream
         (replace (SEQUIN LEAFCACHEDFILE) of CONNECTION with NIL])
        (RETURN)))
    (SETQ LEAFHANDLE (fetch (LEAFSTREAM LEAFHANDLE) of STREAM])
  (COND
    ((EQ (fetch (SEQUIN SEQSTATE) of CONNECTION)
         \SS.OPEN) ; Don't bother sending anything if the connection is already gone
     (SETQ OPUP (ALLOCATE.PUP))
     (SETQ DATA (fetch PUPCONTENTS of OPUP))
     (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.CLOSE \OPCODE.SHIFT))
     (replace (LEAFDATA HANDLE) of DATA with LEAFHANDLE)
     (replace (LEAFDATA LEAFLLENGTH) of DATA with \LEN.CLOSEREQUEST)
     ; Note: don't give the stream to the sequin if we are quietly
     ; closing the cache, because we don't want this to result in a
     ; bogus not responding error
     (\SENDLEAF CONNECTION OPUP (AND (NEQ FORCE :CACHE)
                                     STREAM)
              NIL T)))
    (STREAM ; no good anymore
     (OR INTERNAL (replace (LEAFSTREAM LEAFREALLYOPEN) of STREAM with NIL))
     (replace (LEAFSTREAM LEAFPAGECACHE) of STREAM with NIL)
     (replace (LEAFSTREAM LEAFCONNECTION) of STREAM with NIL]))
  )

```

(\LEAF.DELETEFILE

```

[LAMBDA (FILENAME DEV) ; Edited 2-Nov-92 03:35 by sybalsky:mv:envos
  (PROG ((OPUP (ALLOCATE.PUP))
        (STREAM (\LEAF.GETFILE DEV FILENAME 'OUTPUT 'OLDEST T 'NODATES))
        DATA IPUP)
    (RETURN (COND
      (STREAM (SETQ DATA (fetch PUPCONTENTS of OPUP))
              (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.DELETE \OPCODE.SHIFT))
              (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
              (replace (LEAFDATA LEAFLLENGTH) of DATA with \LEN.CLOSEREQUEST)
              (COND
                ((SETQ IPUP (\SENDLEAF (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
                                       OPUP STREAM))
                 (RELEASE.PUP IPUP)
                 (replace (LEAFSTREAM LEAFCONNECTION) of STREAM with NIL)
                 ; The leaf file connection is now gone
                 (fetch (STREAM FULLFILENAME) of STREAM])
                (T)
              )
            )
    )

```

(\LEAF.DEVICEP

```

[LAMBDA (HOST LEAFDEV) ; Edited 26-Apr-90 11:56 by nm

```

;;; Returns the device corresponding to this HOST, or NIL if it is an illegal leaf host

```

(PROG (NAME DEVICE SEQUIN CONN)
  (RETURN (COND
    ([AND (STRPOS "DSK" HOST 1 NIL T NIL UPPERCHARARRAY)
          (for I from 4 to (NCHARS HOST) always (SMALLP (NTHCHAR HOST I)
                ; Kludge: Name of form DSKn: don't bother
                NIL)
          ((STRPOS '%: HOST)
           ; NS host, skip it. Would be nice to have more orderly name
           ; tests

```

```

NIL)
((AND (EQL \MACHINETYPE \MAIKO)
      (STRPOS "UNIX" HOST 1 NIL T NIL UPPERCASEARRAY))
      ; Maiko uses UNIX as a name of local file system.
NIL)
((NULL (SETQ NAME (\CANONICAL.HOSTNAME HOST)))
NIL)
((NULL LEAFDEV)
NAME)
      ; Called as predicate, don't try to open one
((AND (NEQ NAME HOST)
      (SETQ DEVICE (\GETDEVICEFROMNAME NAME T T)))
DEVICE)
((NULL (SETQ SEQUIN (\OPENLEAFCONNECTION NAME)))
NIL)
((type? SEQUIN SEQUIN)
 [\DEFINEDEVICE NAME (SETQ DEVICE (\MAKE.PMAP.DEVICE (create FDEV
DEVICENAME _ NAME
CLOSEFILE _
(FUNCTION \LEAF.CLOSEFILE)
DELETEFILE _
(FUNCTION \LEAF.DELETEFILE)
GETFILEINFO _
(FUNCTION \LEAF.GETFILEINFO)
OPENFILE _
(FUNCTION \LEAF.OPENFILE)
READPAGES _
(FUNCTION \LEAF.READPAGES)
WRITEPAGES _
(FUNCTION \LEAF.WRITEPAGES)
SETFILEINFO _
(FUNCTION \LEAF.SETFILEINFO)
TRUNCATEFILE _
(FUNCTION \LEAF.TRUNCATEFILE)
GETFILENAME _
(FUNCTION \LEAF.GETFILENAME)
REOPENFILE _
(FUNCTION \LEAF.REOPENFILE)
GENERATEFILES _
(FUNCTION \LEAF.GENERATEFILES)
EVENTFN _
(FUNCTION \LEAF.EVENTFN)
DIRECTORYNAMEP _
(FUNCTION \LEAF.DIRECTORYNAMEP)
HOSTNAMEP _
(FUNCTION NIL)
RENAMEFILE _
(FUNCTION \LEAF.RENAMEFILE)
DEVICEINFO _
(create PUPFILESERVER
PFSNAME _ NAME
PFSOSTYPE _
(GETHOSTINFO
NAME
'OSTYPE)
PFSLEAFSEQUIN _ SEQUIN)
FORCEOUTPUT _
(FUNCTION \LEAF.FORCEOUTPUT)
OPENP _ (FUNCTION
\GENERIC.OPENP)
REGISTERFILE _
(FUNCTION \ADD-OPEN-STREAM)
UNREGISTERFILE _
(FUNCTION
\GENERIC-UNREGISTER-STREAM)
BREAKCONNECTION _
(FUNCTION \LEAF.BREAKCONNECTION
]
DEVICE)
((AND \FTPAVAILABLE (SETQ CONN (\FTP.OPEN.CONNECTION NAME)))
(\RELEASE.FTPCONNECTION CONN)
\FTPDEV])

```

(\LEAF.RECONNECT

; Edited 24-May-91 15:11 by jds

```

[LAMBDA (DEVICE OLDONLY)
(WITH.MONITOR \LEAFCONNECTIONLOCK
[PROG ((INFO (fetch DEVICEINFO of DEVICE))
SEQUIN)
(RETURN (COND
((AND (SETQ SEQUIN (fetch (PUPFILESERVER PFSLEAFSEQUIN) of INFO))
(EQ (fetch (SEQUIN SEQSTATE) of SEQUIN)
\SS.OPEN))
SEQUIN)
([AND (NOT OLDONLY)
(type? SEQUIN (SETQ SEQUIN (\OPENLEAFCONNECTION (fetch (PUPFILESERVER PFSNAME)
of INFO)
(replace (PUPFILESERVER PFSLEAFSEQUIN) of INFO with SEQUIN)

```

SEQUIN]]))

(\LEAF.DIRECTORYNAMEP

[LAMBDA (HOST/DIR DEV)

; Edited 24-May-91 15:11 by jds

;; True if HOST/DIR is a valid host/directory specification, NIL if not. We do this by trying to open an unlikely filename on the dir and see if the error
;; we get is 'file not found' or 'invalid directory'

(LET (INFO)
(COND

((NULL (UNPACKFILENAME.STRING HOST/DIR 'DIRECTORY)) ; No directory field--assume is malformed. Don't do GETFILE
; below, since that packfilename could coerce a non-directory
; into a directory

NIL)
((CL:MEMBER HOST/DIR (fetch (PUPFILESERVER PFSKNOWNDIRS) of (SETQ INFO (fetch DEVICEINFO of DEV)))

:TEST
(if (EQ (fetch (PUPFILESERVER PFSOSTYPE) of INFO)
'UNIX)

then ; Stupid case-sensitive

'CL:STRING= ; We already know this directory is ok
else 'STRING-EQUAL))

T)
((\LEAF.GETFILE DEV (PACKFILENAME.STRING 'DIRECTORY HOST/DIR 'NAME "QXZRYU")

'INPUT
'OLD T 'DIRECTORY)
(push (fetch (PUPFILESERVER PFSKNOWNDIRS) of INFO)
HOST/DIR)

; Returning T tells the caller to canonicalize the host name for me

T])

(\LEAF.GENERATEFILES

[LAMBDA (DEVICE PATTERN DESIREDPROPS OPTIONS)

(\* bvm%: "28-Apr-84 00:02")

(OR (AND \FTPAVAILABLE (\FTP.GENERATEFILES DEVICE PATTERN DESIREDPROPS OPTIONS))
(\GENERATENOFILES DEVICE PATTERN DESIREDPROPS OPTIONS]))

(\LEAF.GETFILE

[LAMBDA (DEVICE FILENAME ACCESS RECOG NOERROR OPTION OLDSTREAM REALLYOPEN)

; Edited 2-Nov-92 03:35 by sybalsky:mv:envos

;;; Opens FILENAME for indicated ACCESS and RECOG, returning a STREAM, optionally smashing DEADSTREAM, on the resulting file, which is now
;;; open. If NOERROR is T, returns NIL on errors; if NOERROR is FIND, returns NIL only on file not found errors. OPTION specifies special way to not
;;; really open the file; choices are --- NAME -- used to get a full file name: in this case, the fullname is returned, and the file is closed on exit ---
;;; DIRECTORY -- FILENAME is a directory specification, not a 'real' filename. Return NIL if the directory doesn't exist, T if it does.

(PROG ((DEVINFO (fetch DEVICEINFO of DEVICE))

CONNECTION MODE FILELENGTH CACHEDSTREAM LEAFHANDLE HOST REMOTENAME NAME/PASS OUTCOME CONNECTNAME/PASS
OPUP IPUP DATA)

(COND

((SETQ HOST (\PARSE.REMOTE.FILENAME FILENAME NOERROR DEVICE))

(SETQ REMOTENAME (CDR HOST))

(SETQ HOST (CAR HOST)))

(T (RETURN)))

(SETQ CONNECTION (fetch (PUPFILESERVER PFSLEAFSEQUIN) of DEVINFO))

TOP (OR CONNECTION (SETQ CONNECTION (\LEAF.RECONNECT DEVICE))

(RETURN))

(COND

([AND (fetch (SEQUIN LEAFCACHEDFILE) of CONNECTION)

(SETQ OUTCOME (WITH.MONITOR (fetch (SEQUIN LEAFOPENCLOSELOCK) of CONNECTION)

[AND (SETQ CACHEDSTREAM (fetch (SEQUIN LEAFCACHEDFILE) of CONNECTION))

(SELECTQ ACCESS

((NONE INPUT)

(COND

((AND (NOT OLDSTREAM)

(EQ (fetch (STREAM FULLFILENAME) of CACHEDSTREAM)

FILENAME)

(COND

((NOT REALLYOPEN)

T)

((fetch (LEAFSTREAM LEAFREALLYOPEN) of

CACHEDSTREAM

; Asking for a new REAL opening of the file, so don't use cache

NIL)

(T (replace (LEAFSTREAM LEAFREALLYOPEN)

of CACHEDSTREAM with T)

T)))

; We already have this file open, and its open state is correct

(SELECTQ OPTION

(NAME FILENAME)

(DATES (\LEAF.GETFILEDATES CACHEDSTREAM)

CACHEDSTREAM)

CACHEDSTREAM)))

(COND

((NOT (fetch (LEAFSTREAM LEAFREALLYOPEN) of CACHEDSTREAM))

; Close the cached file in case it is the one we are now trying to

; open for write

```

(replace (SEQUIN LEAFCACHEDFILE) of CONNECTION with NIL)
(\LEAF.CLOSEFILE CACHEDSTREAM T NIL T)
NIL)] ]
(RETURN OUTCOME))
(SETQ NAME/PASS (\INTERNAL/GETPASSWORD HOST))
RETRY
(SETQ OPUP (ALLOCATE.PUP))
(SETQ DATA (fetch PUPCONTENTS of OPUP))
(\CLEARBYTES DATA 0 \LEN.OPENREQUEST)
(replace (LEAFDATA OPCODE) of DATA with \LEAFOP.OPEN)
(replace (LEAFDATA OPENMODE) of DATA with (+ (SELECTQ ACCESS
((INPUT NONE)
\LEAF.READBIT)
((OUTPUT APPEND BOTH)
(+ \LEAF.WRITEBIT \LEAF.EXTENDBIT))
(LISPERROR "ILLEGAL ARG" ACCESS))
(SELECTQ RECOG
(OLD \LEAF.DEFAULT.HIGHEST)
(OLD/NEW (+ \LEAF.DEFAULT.HIGHEST \LEAF.CREATEBIT))
(NEW (+ \LEAF.DEFAULT.NEXT \LEAF.CREATEBIT))
(OLDEST \LEAF.DEFAULT.LOWEST)
(NIL (SELECTQ ACCESS
(OUTPUT (+ \LEAF.DEFAULT.NEXT \LEAF.CREATEBIT
))
((INPUT NONE)
\LEAF.DEFAULT.HIGHEST)
(+ \LEAF.DEFAULT.HIGHEST \LEAF.CREATEBIT)))
(LISPERROR "ILLEGAL ARG" RECOG))
\LEAF.EXPLICIT.ANY))
(replace (LEAFDATA LEAFLENGTH) of DATA with \LEN.OPENREQUEST)
(\ADDLEAFSTRING OPUP (CAR NAME/PASS))
(\ADDLEAFSTRING OPUP (CDR NAME/PASS)
T)
(\ADDLEAFSTRING OPUP (CAR CONNECTNAME/PASS)) ; Connect name
(\ADDLEAFSTRING OPUP (CDR CONNECTNAME/PASS)
T) ; Connect password
(\ADDLEAFSTRING OPUP REMOTENAME)
[RETURN (COND
((SETQ IPUP (\SENDLEAF CONNECTION OPUP (if (EQ OPTION 'DIRECTORY)
then
; Don't reveal that silly name if connection fails to respond
T
else FILENAME)
T))
(PROG1 [SELECTC (SETQ OUTCOME (fetch (LEAFPACKET LEAFSTATUS) of IPUP))
(\LEAF.GOODSTATUS
(SETQ FILELENGTH (fetch (LEAFDATA FILEADDRESS) of (fetch PUPCONTENTS
of IPUP)))
(SETQ LEAFHANDLE (fetch (LEAFDATA HANDLE) of (fetch PUPCONTENTS of IPUP)))
[COND
((EQ OPTION 'DIRECTORY) ; just wanted to know if directory is valid. Obviously is
(\LEAF.CLOSEFILE NIL CONNECTION LEAFHANDLE)
T)
(T (COND
((NOT (PROG1 OLDSTREAM
(OR OLDSTREAM (SETQ OLDSTREAM
(create STREAM
DEVICE _ DEVICE)))
(replace (LEAFSTREAM LEAFCONNECTION) of OLDSTREAM
with CONNECTION)
(replace (LEAFSTREAM LEAFHANDLE) of OLDSTREAM
with LEAFHANDLE)))
(replace (STREAM FULLFILENAME) of OLDSTREAM
with (OR (\LEAF.READFILENAME OLDSTREAM DEVINFO)
FILENAME)))
(T (replace (LEAFSTREAM LEAFPAGECACHE) of OLDSTREAM with NIL)))
[COND
((EQ ACCESS 'OUTPUT)
; Note: OUTPUT means there is no file to start with! so EOF=0
(replace (STREAM EPAGE) of OLDSTREAM
with (replace (STREAM EOFFSET) of OLDSTREAM with 0)))
(T (replace (STREAM EPAGE) of OLDSTREAM with (fetch (BYTEPTR PAGE)
of FILELENGTH))
(replace (STREAM EOFFSET) of OLDSTREAM
with (fetch (BYTEPTR OFFSET) of FILELENGTH]
(COND
((EQ OPTION 'NAME)
(PROG1 (fetch (STREAM FULLFILENAME) of OLDSTREAM)
(\LEAF.CLOSEFILE OLDSTREAM T)))
(T (COND
((OR (EQ OPTION 'DATES)
(NEQ ACCESS 'NONE))
(\LEAF.GETFILEDATES OLDSTREAM T)))
OLDSTREAM])
(\PASSWORD.ERRORS ; password error
(COND
((SETQ NAME/PASS (\FIXPASSWORD OUTCOME CONNECTION))

```





```

(VMS (LIST "[" DIR "]"))
(LIST "<" DIR ">"))
(LIST NAME)
(if (AND EXT (NEQ 0 (NCHARS EXT)))
  then (LIST "." EXT)
  else (SELECTQ OSTYPE
        ((TENEX TOPS20 VMS)
         ; even extensionless files have a dot
         (LIST "."))
        NIL))
(AND VERSION (NEQ 0 (NCHARS VERSION))
 (LIST (SELECTQ OSTYPE
        (TOPS20 ".")
        (IFS UNIX)
        ; Unix? you ask. Well, the Leaf server doesn't seem to
        ; understand semicolon, even though that's how the files are
        ; stored!
        "!.")
        ";")
VERSION])

```

(\LEAF.STRIP.QUOTES

```

[LAMBDA (NAME) ; Edited 11-Jan-88 16:13 by bvm
  ;; Remove quotes from file NAME, since remote devices never understand our quoting convention (actually, there isn't one in the Leaf protocol).
  ;; Currently, we only remove quotes that look like they're quoting something interesting.
  (CONCATCODES (for (TAIL (CHCON NAME)) by (CDR TAIL) while TAIL
    collect (if (AND (EQ (CAR TAIL)
                        (CHARCODE "'"))
                (CDR TAIL))
              then (SETQ TAIL (CDR TAIL)) ; skip quote
                (CAR TAIL]))

```

(\LEAF.GETFILEDATES

```

[LAMBDA (STREAM FLG) ; Edited 24-May-91 15:07 by jds
  (PROG ((INFOBLK (fetch (LEAFSTREAM LEAFINFO) of STREAM))
        START)
    (COND
      [(NOT INFOBLK)
       (replace (LEAFSTREAM LEAFINFO) of STREAM with (SETQ INFOBLK (create LEAFINFOBLOCK)
       (NOT FLG)
       (RETURN INFOBLK)))
      [(COND
        ((SETQ START (\LEAF.READFILEPROP STREAM 0 (UNFOLD 3 BYTESPERCELL)))
         ; Get 3 info dates from IFS leader
         (\BLT INFOBLK (CDR START)
                     (UNFOLD 3 WORDSPERCELL))
         (RELEASE.PUP (CAR START)))
        (T
         (\CLEARBYTES INFOBLK 0 (UNFOLD 3 BYTESPERCELL)
         (\LEAF.SETVALIDATION STREAM)
         (RETURN INFOBLK]))

```

(\LEAF.GETFILEINFO

```

[LAMBDA (STREAM ATTRIBUTE DEV) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos
  (COND
    ((type? STREAM STREAM) ; Handle open case easily
     (\LEAF.GETFILEINFO.OPEN STREAM ATTRIBUTE))
    (T (PROG (DEVINFO SEQUIN RESULT)
      [COND
        ((FMEMB ATTRIBUTE ' (CREATIONDATE ICREATIONDATE))
         ;; Use the LOOKUPFILE protocol. Would like to have LENGTH here, too, but might disagree with Leaf due to race
         ;; conditions; e.g. LENGTH of a file that I just had closed could get an old length
        (COND
          ((AND [SETQ SEQUIN (fetch (PUPFILESERVER PFSLEAFSEQUIN) of (SETQ DEVINFO
                                                                    (fetch DEVICEINFO of DEV)
                                                                    (SETQ RESULT (fetch (SEQUIN LEAFACHEDFILE) of SEQUIN))
                                                                    (EQ (fetch (STREAM FULLFILENAME) of RESULT)
                                                                    STREAM)) ; A name we know about
              (RETURN (\LEAF.GETFILEINFO.OPEN RESULT ATTRIBUTE)))
           ((NEQ (SETQ RESULT (\IFS.LOOKUPFILE STREAM 'OLD ATTRIBUTE DEVINFO))
                '?))
            (RETURN RESULT]
         ;; To get attributes, have to open file, read them, then close.
        (RETURN (COND
          ((SETQ STREAM (\LEAF.GETFILE DEV STREAM 'NONE 'OLD))
           (PROG1 (\LEAF.GETFILEINFO.OPEN STREAM ATTRIBUTE)
                 (\LEAF.CLOSEFILE STREAM T]))

```

(\LEAF.GETFILEINFO.OPEN

```

[LAMBDA (STREAM ATTRIBUTE) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos
  (SELECTQ ATTRIBUTE
    (LENGTH (create BYTEPTR
      PAGE _ (fetch (STREAM EPAGE) of STREAM)
      OFFSET (fetch (STREAM EOFSET) of STREAM)))
    (CREATIONDATE (GDATE (\LEAF.GETFILEINFO.OPEN STREAM 'ICREATIONDATE)))
    (WRITEDATE (GDATE (\LEAF.GETFILEINFO.OPEN STREAM 'IWRITEDATE)))
    (READDATE (GDATE (\LEAF.GETFILEINFO.OPEN STREAM 'IREADDATE)))
    (ICREATIONDATE
      (ALTO.TO.LISP.DATE (fetch (LEAFINFOBLOCK LFCREATIONDATE) of (\LEAF.GETFILEDATES STREAM))))
    (IWRITEDATE (ALTO.TO.LISP.DATE (fetch (LEAFINFOBLOCK LFWRTEDATE) of (\LEAF.GETFILEDATES STREAM))))
    (IREADDATE (ALTO.TO.LISP.DATE (fetch (LEAFINFOBLOCK LFREADDATE) of (\LEAF.GETFILEDATES STREAM))))
    ((TYPE BYTESIZE)
      [PROG (FT (BYTESIZE 8))
        [SETQ FT (COND
          [(SETQ FT (\LEAF.READFILEPROP STREAM \OFFSET.FILETYPE \LEN.FILETYPE&SIZE))
            ; FT = (pup .base)
            (PROG1 (SELECTC (\GETBASE (CDR FT)
              0)
                (\FT.UNKNOWN NIL)
                (\FT.TEXT 'TEXT)
                (\FT.BINARY (SETQ BYTESIZE (\GETBASE (CDR FT)
                  1))
                  'BINARY)
                  '?))
              (RELEASE.PUP (CAR FT)))]
          (T '?)]
        (RETURN (COND
          ((EQ ATTRIBUTE 'BYTESIZE)
            BYTESIZE)
          (T FT]))
        (AUTHOR [LET ((BASE (\LEAF.READFILEPROP STREAM \OFFSET.AUTHOR \LEN.AUTHOR))
          (AND BASE (PROG1 (GetBcplString (CDR BASE))
            (RELEASE.PUP (CAR BASE)))))]
        ((BACKUPDATE IBACKUPDATE)
          [LET ((BASE (\LEAF.READFILEPROP STREAM \OFFSET.BACKUPDATE \LEN.DATE))
            DT)
            (COND
              (BASE (SETQ DT (fetch (LEAFINFOBLOCK LFCREATIONDATE) of (CDR BASE)))
                (RELEASE.PUP (CAR BASE))
                (if (NEQ DT 0)
                  then ; Zero means it hasn't been
                    (SETQ DT (ALTO.TO.LISP.DATE DT))
                    (if (EQ ATTRIBUTE 'IBACKUPDATE)
                      then DT
                      else (GDATE DT)))
                NIL])
          NIL])

```

(\LEAF.GETFILENAME

```

[LAMBDA (NAME RECOG DEV) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos
  (PROG ((DEVINFO (fetch DEVICEINFO of DEV))
    SEQUIN RESULT)
    (RETURN (OR [COND
      ((AND (SETQ SEQUIN (fetch (PUPFILESERVER PFSLEAFSEQUIN) of DEVINFO))
        (SETQ RESULT (fetch (SEQUIN LEAFCACHEDFILE) of SEQUIN))
        (EQ (fetch (STREAM FULLFILENAME) of RESULT)
          NAME))
          ; A name we know about
          NAME)
      ((AND (NEQ RECOG 'NEW)
        (NEQ (SETQ RESULT (\IFS.LOOKUPFILE NAME RECOG 'NAME DEVINFO))
          '?))
        RESULT)
      (T (\LEAF.GETFILE DEV NAME 'NONE RECOG T 'NAME]
    (SELECTQ RECOG
      ((NEW OLD/NEW)
        (\GENERIC.OUTFILEP NAME DEV))
      NIL])

```

(\LEAF.OPENFILE

```

[LAMBDA (FILENAME ACCESS RECOG OTHERINFO DEV) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos
  (PROG ((DEVINFO (fetch DEVICEINFO of DEV))
    STREAM TYPE BYTESIZE OLDHANDLE CRDATE PROPS SEQUIN EOL)
    [COND
      ((type? STREAM FILENAME) ; Hmm? trying to reopen, perhaps?
        (COND
          ((fetch (STREAM ACCESS) of FILENAME)
            (RETURN (LISPERROR "FILE WON'T OPEN" FILENAME)))
          (T (SETQ FILENAME (fetch (STREAM FULLFILENAME) of (SETQ OLDHANDLE FILENAME]
        (for X in OTHERINFO do ; Check device-dependent parameters
          (SELECTQ [CAR (OR (LISTP X)
            (SETQ X (LIST X T]
            ((TYPE FILETYPE) ; Set the file TYPE (TEXT or BINARY)
              (SETQ TYPE (CDR X)))
            (BYTESIZE (SETQ BYTESIZE (OR (FIXP (CADR X))
              (\ILLEGAL.ARG X))))

```

```

(CREATIONDATE (SETQ CRDATE (IDATE (CADR X))))
(ICREATIONDATE
  (SETQ CRDATE (OR (FIXP (CADR X))
                   (\ILLEGAL.ARG X))))
(DON'T.CHANGE.DATE
 ;; Don't change create date. In order to do this, we have to look at the current date of the file, save it,
 ;; then rewrite when we open the file for real
(COND
  ((AND (NEQ ACCESS 'INPUT)
        (SETQ OLDHANDLE (\LEAF.GETFILE DEV FILENAME 'NONE 'OLD T
                                     'DATES OLDHANDLE)))
   (SETQ FILENAME (fetch (STREAM FULLFILENAME) of OLDHANDLE))
   (SETQ CRDATE (\LEAF.CREATIONDATE OLDHANDLE))
   (\LEAF.CLOSEFILE OLDHANDLE NIL NIL T)))
 (SEQUENTIAL ;Hook for FTP
  (COND
   ((AND (CADR X)
          \FTPAVAILABLE
          (OR (NEQ (fetch (PUPFILESERVER PFSOSTYPE) of DEVINFO)
                'UNIX)
              UNIXFTPFLG)
          (SETQ STREAM (\FTP.OPENFILE FILENAME ACCESS RECOG
                                     OTHERINFO)))
    (RETURN))))
 (EOL (SETQ EOL (SELECTQ (CADR X)
                        (CR CR.EOLC)
                        (LF LF.EOLC)
                        (CRLF CRLF.EOLC)
                        (\ILLEGAL.ARG X))))
 (push PROPS X)))
[COND
 (STREAM)
 ((SETQ STREAM (\LEAF.GETFILE DEV FILENAME ACCESS RECOG 'FIND NIL OLDHANDLE T))
  ; Returns NIL if file not found
 (COND
  (CRDATE (\LEAF.SETCREATIONDATE STREAM CRDATE))
  (T (\LEAF.GETFILEDATES STREAM)))
 (COND
  ([AND (NEQ ACCESS 'INPUT)
        (COND
         (TYPE ; Type NIL overrides default
          (SETQ TYPE (CAR TYPE)))
          (T (AND (SETQ TYPE DEFAULTFILETYPE)
                  (EQ (fetch (STREAM EPAGE) of STREAM)
                      0)
                  (EQ (fetch (STREAM EOFFSET) of STREAM)
                      0))
           ;; Set file type if explicitly requested, or if this is a new output file and there is a global default
           (\LEAF.SETFILETYPE STREAM TYPE BYTESIZE)))
   (SETQ SEQUIN (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM))
   (COND
    ((IGREATERP (fetch (SEQUIN LEAFCACHEHITS) of SEQUIN)
                32767)
     ; Keep counters from overflowing
     (replace (SEQUIN LEAFCACHEHITS) of SEQUIN with 0)
     (replace (SEQUIN LEAFCACHEMISSES) of SEQUIN with 0)))
    (COND
     ((IGREATERP (fetch (SEQUIN %#SEQTIMEOUTS) of SEQUIN)
                 32767)
      (replace (SEQUIN %#SEQRESTARTS) of SEQUIN with 0)
      (replace (SEQUIN %#SEQTIMEOUTS) of SEQUIN with 0)
      (replace (SEQUIN %#SEQDUPLICATES) of SEQUIN with 0)))
     (replace (STREAM CBUFSIZE) of STREAM with 0) ; For the benefit of uCode and PageMapped fns
     (replace (STREAM CBUFPTR) of STREAM with NIL)
     (replace (STREAM EOLCONVENTION) of STREAM with (OR EOL (SELECTQ (fetch (PUPFILESERVER PFSOSTYPE)
                                     of DEVINFO)
                             (TENEX TOPS20)
                             CRLF.EOLC)
                             (UNIX LF.EOLC)
                             CR.EOLC]
   (RETURN STREAM]))

```

(\LEAF.READFILENAME

```

[LAMBDA (STREAM DEVINFO) ; Edited 24-May-91 15:11 by jds
 (LET ([RE MOTENAME (LET ((NAMEBASE (\LEAF.READFILEPROP STREAM \OFFSET.FILENAME \MAXLEN.FILENAME)))
                        ; Returns (pup . base)
                        (AND NAMEBASE (PROG1 (GetBcplString (CDR NAMEBASE))
                                                (RELEASE.PUP (CAR NAMEBASE))))]
      (OSTYPE (fetch (PUPFILESERVER PFSOSTYPE) of DEVINFO)))
 (COND
  ((NOT REMOTENAME) ; Some hosts may refuse us the name
   NIL)
  (T [SETQ REMOTENAME (CL:APPLY (FUNCTION PACKFILENAME.STRING)
                                'HOST
                                (fetch (SEQUIN SEQNAME) of (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM))

```

```
(UNPACKFILENAME.STRING (\LEAF.ADD.QUOTES REMOTENAME 'IFS)
  NIL NIL (if (EQ OSTYPE 'UNIX)
              then ; Kludge: call it an IFS, since current Unix servers return ! for the
                  ; version.
                  ' IFS
              else OSTYPE])

(if *UPPER-CASE-FILE-NAMES*
  then (MKATOM (U-CASE REMOTENAME))
  else REMOTENAME])
```

**(\LEAF.ADD.QUOTES**

[LAMBDA (NAME OSTYPE) ; Edited 11-Jan-88 16:32 by bvm

;; The only funny char we know about is quote, so quote all the quotes with a quote.

```
(bind (N _ 1)
  I PIECES while (SETQ I (STRPOS "\"" NAME N)) do (push PIECES "\"" (SUBSTRING NAME N I))
              (SETQ N (ADD1 I))

  finally (if (AND (EQ OSTYPE 'IFS)
                  (SETQ I (STRPOS "!" NAME N)))
            then ; Yet another piece of nonsense: for IFS file ending in dot, we'd
                ; better quote the dot, lest it be discarded
                (push PIECES "\"" (SUBSTRING NAME N (SUB1 I)))
              (SETQ N I))
  (RETURN (if PIECES
            then (if (<= N (NCHARS NAME))
                    then (push PIECES (SUBSTRING NAME N))
                    (CONCATLIST (DREVERSE PIECES)))
            else ; nothing got quoted
              NAME]))
```

**(\LEAF.READFILEPROP**

[LAMBDA (STREAM OFFSET LEN) ; Edited 24-May-91 15:07 by jds

;; Read a chunk of the IFS leader page starting at OFFSET for LEN bytes. Returns a dotted pair, car of which is the reply pup and CDR is a pointer  
;; inside it to the desired data

```
(PROG ((CONNECTION (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM))
      (OPUP (ALLOCATE.PUP))
      DATA IPUP)
  (SETQ DATA (fetch PUPCONTENTS of OPUP))
  (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.READ \OPCODE.SHIFT))
  (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
  (replace (LEAFDATA FILEADDRESS) of DATA with (IDIFFERENCE OFFSET \BYTES.PER.TRIDENT.PAGE))
  (replace (LEAFDATA SIGNEXTEND) of DATA with 0)
  (replace (LEAFDATA DATALENGTH) of DATA with LEN)
  (replace (LEAFDATA LEAFLLENGTH) of DATA with \LEN.FILEREQUEST)
  (SETQ IPUP (\SENDLEAF CONNECTION OPUP STREAM NOFILEPROPERROR))
  (RETURN (COND
    ((EQ (fetch (LEAFPACKET LEAFSTATUS) of IPUP)
         \LEAF.GOODSTATUS)
     (CONS IPUP (\ADDBASE (fetch PUPCONTENTS of IPUP)
                          (FOLDLO \LEN.READANSWER BYTESPERWORD]))
```

**(\LEAF.READPAGES**

[LAMBDA (STREAM FIRSTPAGE BUFFERLIST) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos

```
(for BUF inside BUFFERLIST as PAGE# from FIRSTPAGE bind LEN
  sum [COND
    ((.PAGE.IS.AFTER.EOF. STREAM PAGE#) ; after end of file
     (SETQ LEN 0))
    (T (PROG (OPUP IPUP DATA)
             RETRY
             (SETQ OPUP (\LEAF.REQUESTPAGE STREAM PAGE# T))
             (for NEWPAGE# from (ADD1 PAGE#) as I to (fetch (LEAFSTREAM LEAFCACHECNT) of STREAM)
              until (.PAGE.IS.AFTER.EOF. STREAM NEWPAGE#) do
                ; Ask for pages immediately following this one, too
                (\LEAF.REQUESTPAGE STREAM NEWPAGE#))
             (until (NEQ (SETQ IPUP (fetch EPUSERFIELD of OPUP))
                        STREAM)
                    do (AWAIT.EVENT [fetch (SEQUIN SEQEVENT) of (OR (fetch (LEAFSTREAM LEAFCONNECTION)
                                                                              of STREAM)
                                                                      (LISPERROR "FILE NOT OPEN"
                                                                      (fetch (STREAM FULLFILENAME)
                                                                      of STREAM]
                                                                      \ETHERTIMEOUT))
                    (RELEASE.PUP OPUP)
                    (COND
                     ((AND (NEQ IPUP \LEAF.BROKEN.STATUS)
                           (NEQ (fetch (LEAFDATA LEAFOPCODE) of (SETQ DATA (fetch PUPCONTENTS of IPUP)))
                                \LEAFOP.ERROR))
                      (SETQ LEN (- (fetch (LEAFDATA LEAFLLENGTH) of DATA)
                                   \LEN.READANSWER))
                      (\BLT BUF (\ADDBASE DATA (FOLDLO \LEN.READANSWER BYTESPERWORD))
                               (FOLDHI LEN BYTESPERWORD))
                      (RELEASE.PUP IPUP)
                      (RETURN LEN))
```

```

((NOT (READABLE STREAM))
 (LISPERROR "FILE NOT OPEN" (fetch (STREAM FULLFILENAME) of STREAM)))
(NEQ IPUP \LEAF.BROKEN.STATUS)
(\LEAF.ERROR IPUP (fetch (STREAM FULLFILENAME) of STREAM)
 (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
 OPOP))
(T (HELP "Failed to read page of file" (fetch (STREAM FULLFILENAME) of STREAM))
 (GO RETRY))
[COND
 (((< LEN BYTESPERPAGE)
 (\CLEARBYTES BUF LEN (- BYTESPERPAGE LEN)
 LEN))

```

(\LEAF.REQUESTPAGE

```

[LAMBDA (STREAM PAGE# IMMEDIATE) ; Edited 24-May-91 15:07 by jds
 ;; Requests PAGE# of STREAM, possibly finding it in the cache first. If IMMEDIATE is true, then we want the page now, and it should be removed
 ;; from the cache and returned; otherwise it is completely optional whether we ask for the page at all or what we return
 (PROG ((CACHE (\LEAF.LOOKUPCACHE STREAM PAGE# IMMEDIATE))
 OPOP DATA)
 [COND
 ((CDR CACHE) ; Cache hit!
 [COND
 (IMMEDIATE (INCLEAFSTAT (fetch (SEQUIN LEAFCACHEHITS) of (fetch (LEAFSTREAM LEAFCONNECTION)
 of STREAM)))
 (COND
 ((ILESSP (fetch (LEAFSTREAM LEAFCACHECNT) of STREAM)
 \LEAF.MAXLOOKAHEAD) ; Reward STREAM for being sequential
 (add (fetch (LEAFSTREAM LEAFCACHECNT) of STREAM)
 1)
 (RETURN (CDR CACHE)
 [COND
 (IMMEDIATE ; Cache miss, so we probably aren't very sequential; be more
 ; cautious
 (replace (LEAFSTREAM LEAFCACHECNT) of STREAM with 1)
 (INCLEAFSTAT (fetch (SEQUIN LEAFCACHEMISSES) of (fetch (LEAFSTREAM LEAFCONNECTION)
 of STREAM)
 [SETQ DATA (fetch PUPCONTENTS of (SETQ OPOP (ALLOCATE.PUP)
 (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.READ \OPCODE.SHIFT))
 (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
 (replace (LEAFDATA FILEADDRESS) of DATA with (create BYTEPTR
 PAGE _ PAGE#
 OFFSET _ 0))
 (replace (LEAFDATA READWRITEMODE) of DATA with \LEAFMODE.DONTEXTEND) ; i.e. don't attempt to read past EOF, in case this is the last page
 (replace (LEAFDATA DATALENGTH) of DATA with BYTESPERPAGE)
 (replace (LEAFDATA LEAFLLENGTH) of DATA with \LEN.FILEREQUEST)
 (RETURN (COND
 ((\SENDLEAF (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
 OPOP STREAM T 'GO (NOT IMMEDIATE))
 (AND CACHE (RPLACD CACHE OPOP))
 OPOP]))

```

(\LEAF.LOOKUPCACHE

```

[LAMBDA (STREAM PAGE# DELETE) ; Edited 24-May-91 15:07 by jds
 ;; Looks up PAGE# in STREAM's cache. If it finds an entry, it returns it and, if DELETE is true, deletes it from the cache; otherwise if DELETE is
 ;; NIL, it inserts a new empty entry for PAGE#
 (for I from 0 bind (CACHE _ (fetch (LEAFSTREAM LEAFPAGECACHE) of STREAM))
 PREV
 while CACHE do [COND
 ((IEQP (CAAR CACHE)
 PAGE#)
 [COND
 ((NOT DELETE) ; Don't remove entry from cache
 )
 (PREV (RPLACD PREV (CDR CACHE)))
 (T (replace (LEAFSTREAM LEAFPAGECACHE) of STREAM with (CDR CACHE)
 (RETURN (CAR CACHE)
 (SETQ CACHE (CDR (SETQ PREV CACHE)))
 finally [COND
 ((NOT DELETE)
 (SETQ CACHE (LIST (CONS PAGE# NIL)))
 (COND
 [PREV (RPLACD PREV CACHE)
 (COND
 ((IGREATERP I \LEAF.MAXCACHE) ; Throw out old cache entries
 (replace (LEAFSTREAM LEAFPAGECACHE) of STREAM with (CDR (fetch (LEAFSTREAM
 LEAFPAGECACHE)
 of STREAM)
 (T (replace (LEAFSTREAM LEAFPAGECACHE) of STREAM with CACHE]
 (RETURN (CAR CACHE]))

```

**(CLEAR.LEAF.CACHE**

; Edited 24-May-91 15:11 by jds

```
[LAMBDA (HOST)
(COND
  (HOST (PROG ([DEVICE (OR (\GETDEVICEFROMNAME HOST T T)
    (AND (SETQ HOST (\CANONICAL.HOSTNAME HOST))
      (\GETDEVICEFROMNAME HOST T T)
    CONNECTION DEVINFO)
  (RETURN (COND
    ((AND DEVICE (type? PUPFILESERVER (SETQ DEVINFO (fetch DEVICEINFO of DEVICE)))
      (SETQ CONNECTION (ffetch (PUPFILESERVER PFSLEAFSEQUIN) of DEVINFO))
      (fetch (SEQUIN LEAFCACHEDFILE) of CONNECTION))
    (\LEAF.FLUSH.CACHE CONNECTION]))
```

**(LEAF.ASSURE.FINISHED**

; Edited 2-Nov-92 03:36 by sybalsky:mv:envos

```
[LAMBDA (STREAM)
(PROG [(SEQUIN (fetch (LEAFSTREAM LEAFCONNECTION) of (SETQ STREAM (\DTEST STREAM 'STREAM)
TOP [COND
  ((type? SEQUIN SEQUIN)
  (WITH.MONITOR (fetch (SEQUIN SEQLOCK) of SEQUIN)
    (bind PUP until [AND [OR [NOT (SETQ PUP (fetch SYSQUEUEHEAD of (fetch (SEQUIN SEQDONEQ)
      of SEQUIN)
      (while PUP never (PROG1 (EQ (fetch EPUSERFIELD of PUP)
        STREAM)
        (SETQ PUP (fetch EPLINK of PUP)))]
      (OR [NOT (SETQ PUP (fetch SYSQUEUEHEAD of (fetch (SEQUIN SEQRETRANSMITQ)
        of SEQUIN)
        (while PUP never (PROG1 (EQ (fetch EPUSERFIELD of PUP)
          STREAM)
          (SETQ PUP (fetch EPLINK of PUP)))]
      do
        ; Not quite right, because it doesn't catch stuff in the retransmit
        ; queue
        (MONITOR.AWAIT.EVENT (fetch (SEQUIN SEQLOCK) of SEQUIN)
          (fetch (SEQUIN SEQEVENT) of SEQUIN)
          \ETHERTIMEOUT))
    (COND
      ((NEQ (fetch (LEAFSTREAM LEAFERRORCNT) of STREAM)
        0)
        (ERROR "Waiting for operation on broken file to finish" (fetch (STREAM FULLFILENAME)
          of STREAM))
      (GO TOP)
    (RETURN T])
```

**(\LEAF.FORCEOUTPUT**

(\* bvm%: "11-Jul-84 11:31")

```
[LAMBDA (STREAM)
  (\PAGED.FORCEOUTPUT STREAM)
  (\LEAF.ASSURE.FINISHED STREAM)]
```

**(\LEAF.FLUSH.CACHE**

; Edited 24-May-91 15:07 by jds

```
[LAMBDA (SEQUIN)
  (WITH.MONITOR (fetch (SEQUIN LEAFOPENCLOSELOCK) of SEQUIN)
  [LET ((CACHE (fetch (SEQUIN LEAFCACHEDFILE) of SEQUIN))
  (COND
    ((NULL CACHE)
     NIL)
    ((fetch (LEAFSTREAM LEAFREALLYOPEN) of CACHE)
     (replace (SEQUIN LEAFCACHEDFILE) of SEQUIN with NIL)
     NIL)
    (T (\LEAF.CLOSEFILE CACHE SEQUIN NIL :CACHE)
     (fetch (SEQUIN SEQNAME) of SEQUIN)))]
```

**(\LEAF.RENAMEFILE**

(\* hdj " 8-May-86 15:20")

```
[LAMBDA (OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE)
  (OR (AND \FTPAVAILABLE (OR (NEQ (GETHOSTINFO (fetch (FDEV DEVICENAME) of OLD-DEVICE)
    'OSTYPE)
    'UNIX)
    UNIXFTPFLG)
  (\FTP.RENAMEFILE OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE))
  (\GENERIC.RENAMEFILE OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE))
```

**(\LEAF.REOPENFILE**

; Edited 2-Nov-92 03:36 by sybalsky:mv:envos

```
[LAMBDA (NAME ACCESS RECOG OTHERINFO FDEV STREAM)
```

::: Called after, say, a LOGOUT to restore the file to its old state. We reopen the file and return a new file handle

```
(PROG (NEWSTREAM OLDINFO NEWINFO OLDDATES)
[COND
  ((NEQ ACCESS 'INPUT)
  ;; Problem: when we reopen the file for write, we change the write and creation dates, so our caller thinks the file has been modified.
  ;; So first open the file for read and look at the dates, and if they're the same as the old filehandle's, prepare to restore them
  (COND
    ((SETQ NEWSTREAM (\LEAF.GETFILE FDEV NAME 'NONE 'OLD T 'DATES))
```

```

[COND
  ((AND [IEQP (fetch (LEAFINFOBLOCK LFCREATIONDATE) of (SETQ OLDINFO (fetch (LEAFSTREAM
    LEAFINFO)
    of STREAM)))
    (fetch (LEAFINFOBLOCK LFCREATIONDATE) of (SETQ NEWINFO (fetch (LEAFSTREAM
    LEAFINFO)
    of NEWSTREAM]
    (IEQP (fetch (LEAFINFOBLOCK LFWRITE DATE) of OLDINFO)
    (fetch (LEAFINFOBLOCK LFWRITE DATE) of NEWINFO)))
    ; Creation and write dates are indeed the same
    (SETQ OLDDATES (\LEAF.CREATIONDATE NEWSTREAM]
  (\LEAF.CLOSEFILE NEWSTREAM NIL NIL T))
  (T ; If we can't even find the file, there's no hope
  (RETURN NIL])
[COND
  ((AND (SETQ NEWSTREAM (\LEAF.GETFILE FDEV NAME ACCESS RECOG T NIL NEWSTREAM))
  OLDDATES) ; Change the file dates to the old dates
  (\LEAF.SETCREATIONDATE NEWSTREAM OLDDATES)
  ;; And smash the validation of the old handle to be the new validation. This is sort of a cheat, but it works to fool \REVALIDATEFILE
  (replace (STREAM VALIDATION) of STREAM with (fetch (STREAM VALIDATION) of NEWSTREAM])
  (RETURN NEWSTREAM])

```

(\LEAF.CREATIONDATE

```

[LAMBDA (STREAM) ; Edited 24-May-91 15:08 by jds
  (ALTO.TO.LISP.DATE (fetch (LEAFINFOBLOCK LFCREATIONDATE) of (fetch (LEAFSTREAM LEAFINFO) of STREAM)])

```

(\LEAF.SETCREATIONDATE

```

[LAMBDA (STREAM DATE) ; Edited 24-May-91 15:08 by jds
  ; DATE is integer in Lisp date format
  (PROG ((INFOBLK (\LEAF.GETFILEDATES STREAM))
  (FILEDATE (LISP.TO.ALTO.DATE DATE))
  (OPUP (ALLOCATE.PUP))
  DATA)
  (SETQ DATA (fetch PUPCONTENTS of OPUP))
  (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.WRITE \OPCODE.SHIFT))
  (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
  (replace (LEAFDATA FILEADDRESS) of DATA with (IDIFFERENCE 0 \BYTES.PER.TRIDENT.PAGE))
  ; negative address into leader page
  (replace (LEAFDATA SIGNEXTEND) of DATA with 0)
  (replace (LEAFDATA DATALENGTH) of DATA with \LEN.DATE)
  (replace (LEAFDATA LEAFFILEDATE) of DATA with FILEDATE)
  (replace (LEAFDATA LEAFLENGTH) of DATA with (IPLUS \LEN.FILEREQUEST \LEN.DATE))
  (\SENDLEAF (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
  OPUP STREAM NIL T)
  (replace (LEAFINFOBLOCK LFCREATIONDATE) of INFOBLK with FILEDATE)
  (\LEAF.SETVALIDATION STREAM) ; Since validation depends on file dates
  (RETURN T])

```

(\LEAF.SETFILEINFO

```

[LAMBDA (STREAM ATTRIBUTE VALUE DEV) (* bvm%: "12-SEP-83 14:16")
  (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 (\LEAF.GETFILE DEV STREAM 'NONE 'OLD]
  (PROG1 (SELECTQ ATTRIBUTE
  (TYPE (\LEAF.SETFILETYPE STREAM VALUE))
  (\LEAF.SETCREATIONDATE STREAM VALUE))
  (COND
  ((NOT WASOPEN)
  (\LEAF.CLOSEFILE STREAM T))))))

```

(\LEAF.SETFILETYPE

```

[LAMBDA (STREAM TYPE BYTESIZE) ; Edited 24-May-91 15:08 by jds
  ; Sets 'type' of file to TEXT or BINARY
  (PROG ((OPUP (ALLOCATE.PUP))
  DATA)
  (SETQ DATA (fetch PUPCONTENTS of OPUP))
  (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.WRITE \OPCODE.SHIFT))
  (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
  (replace (LEAFDATA FILEADDRESS) of DATA with (IDIFFERENCE \OFFSET.FILETYPE \BYTES.PER.TRIDENT.PAGE))
  ; negative address into leader page
  (replace (LEAFDATA SIGNEXTEND) of DATA with 0)
  (replace (LEAFDATA DATALENGTH) of DATA with \LEN.FILETYPE&SIZE)
  ; Patch: IFS code has bug that only lets me do a write with
  ; length=4 here

```

```

[COND
  ((LISTP TYPE) ; E.g. (BINARY 16). Does anyone else know about this?
   (SETQ BYTESIZE (FIXP (CADR TYPE)))
   (SETQ TYPE (CAR TYPE))
  (replace (LEAFDATA LEAFFILETYPE) of DATA with (SELECTQ TYPE
                                                    (TEXT \FT.TEXT)
                                                    (NIL \FT.UNKNOWN)
                                                    \FT.BINARY))
  (replace (LEAFDATA LEAFBYTESIZE) of DATA with (OR BYTESIZE 8))
  (replace (LEAFDATA LEAFLLENGTH) of DATA with (IPLUS \LEN.FILEREQUEST \LEN.FILETYPE&SIZE))
  (\SENDLEAF (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
             OPOP STREAM NIL T)
  (RETURN TYPE])

```

(\LEAF.SETVALIDATION

```

[LAMBDA (STREAM) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos

```

;;; Set the VALIDATION field of STREAM based on the file's write and creation dates

```

(replace (STREAM VALIDATION) of STREAM with (\MAKENUMBER (fetch (LEAFINFOBLOCK LOCREATE)
                                                                of (fetch (LEAFSTREAM LEAFINFO) of STREAM))
                                             (fetch (LEAFINFOBLOCK LOWRITE) of (fetch (LEAFSTREAM LEAFINFO)
                                                                                       of STREAM]))

```

(\LEAF.TRUNCATEFILE

```

[LAMBDA (STREAM LASTPAGE LASTOFF) ; Edited 24-May-91 15:08 by jds

```

;;; Truncate file by doing a zero-length write with the EOF bit set

```

(COND
  (LASTPAGE ; Don't bother if defaulting, we have already set correct length if
            ; so
    (PROG ((OPUP (ALLOCATE.PUP))
           DATA)
          (SETQ DATA (fetch PUPCONTENTS of OPUP))
          (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.WRITE \OPCODE.SHIFT))
          (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
          (replace (LEAFDATA FILEADDRESS) of DATA with (create BYTEPTR
                                                             PAGE _ LASTPAGE
                                                             OFFSET _ LASTOFF))

          (replace (LEAFDATA EOFBIT) of DATA with 1)
          (replace (LEAFDATA DATALENGTH) of DATA with 0)
          (replace (LEAFDATA LEAFLLENGTH) of DATA with \LEN.FILEREQUEST)
          (\SENDLEAF (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
                    OPOP STREAM NIL T)
          (RETURN STREAM])

```

(\LEAF.WRITEPAGES

```

[LAMBDA (STREAM FIRSTPAGE BUFFERLIST) ; Edited 2-Nov-92 03:36 by sybalsky:mv:envos

```

```

(COND
  ((fetch (STREAM 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
  (\LEAF.SETCREATIONDATE STREAM (IDATE))
  (replace (STREAM REVALIDATEFLG) of STREAM with NIL)))

```

(for BUF inside BUFFERLIST as PAGE# from FIRSTPAGE do (\LEAF.LOOKUPCACHE STREAM PAGE# T) ; Invalidate any read-ahead of this page

```

(PROG ((OPUP (ALLOCATE.PUP))
       DATA LEN)
      (SETQ DATA (fetch PUPCONTENTS of OPUP))
      (replace (LEAFDATA OPWORD) of DATA with (LLSH \LEAFOP.WRITE \OPCODE.SHIFT))
      (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
      (replace (LEAFDATA FILEADDRESS) of DATA with (create BYTEPTR
                                                           PAGE _ PAGE#
                                                           OFFSET _ 0))

      [replace (LEAFDATA DATALENGTH) of DATA with (SETQ LEN (COND
                                                         ((NEQ PAGE# (fetch (STREAM EPAGE)
                                                                    of STREAM))
                                                          BYTESPERPAGE)
                                                         (T

```

; On last page, only write as much as we really have (replace (LEAFDATA EOFBIT) of DATA with 1) (fetch (STREAM EOFFSET) of STREAM]

```

(\BLT (\ADDBASE DATA (FOLDLO \LEN.FILEREQUEST BYTESPERWORD))
      BUF
      (FOLDHI LEN BYTESPERWORD))
  (replace (LEAFDATA LEAFLLENGTH) of DATA with (IPLUS \LEN.FILEREQUEST LEN))
  (\SENDLEAF (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
             OPOP STREAM NIL T])

```

)



:: Main routing point for LEAF pups

(DEFINEQ

(\SENDLEAF

; Edited 2-Nov-92 03:36 by sybalsky:mv:envos

```

[LAMBDA (SEQUIN PUP FILENAME NOERROR NOREPLY DONTWAIT)
  (PROG (RESULT)
    TOP (OR SEQUIN (RETURN (LISPERROR "FILE NOT OPEN" FILENAME)))
    (COND
      ((AND (type? STREAM FILENAME)
            (NEQ (fetch (LEAFSTREAM LEAFERRORCNT) of FILENAME)
                 0))
        (ERROR "Attempt to operate on broken file. Do not proceed until the problem has been resolved."
              (fetch (STREAM FULLFILENAME) of FILENAME)))
      (GO TOP)))
  (replace EPUSERFIELD of PUP with FILENAME)
  [replace (LEAFPACKET LEAFFLAGS) of PUP with (LOGOR (COND
    (NOERROR \LF.ALLOWERRORS)
    (T 0))
    (COND
      ((EQ NOREPLY T)
       0)
      (T \LF.WANTANSWER)
    )
  )
  (replace PUPLENGTH of PUP with (IPLUS (fetch (LEAFDATA LEAFLENGTH) of (fetch PUPCONTENTS of PUP))
    \PUPOVLEN))
  (RETURN (COND
    ((NULL (PUTSEQUIN SEQUIN PUP DONTWAIT))
     NIL)
    (NOREPLY T)
    (T (until (NEQ (fetch EPUSERFIELD of PUP)
                  FILENAME)
              do (AWAIT.EVENT (fetch (SEQUIN SEQEVENT) of SEQUIN)
                             \ETHERTIMEOUT))
        (SETQ RESULT (fetch EPUSERFIELD of PUP))
        (COND
          ((EQ RESULT \LEAF.BROKEN.STATUS)
           PUP)
          (T (replace (LEAFPACKET LEAFSTATUS) of RESULT
                    with (COND
                      ((EQ (fetch (LEAFDATA LEAFOPCODE) of (fetch PUPCONTENTS of RESULT))
                          \LEAFOP.ERROR)
                       (fetch (LEAFERRORDATA LEAFERRORCODE) of (fetch PUPCONTENTS of RESULT)))
                      (T \LEAF.GOODSTATUS)))
              (RELEASE.PUP PUP)
              RESULT]))
    )
  )

```

)

:: Managing LEAF connections

(DEFINEQ

(\OPENLEAFCONNECTION

; Edited 24-May-91 15:04 by jds

```

[LAMBDA (HOST)
  (PROG (PROTOCOLS IFSPORT NAME/PASS)
    [COND
      ([OR (MEMB HOST NONLEAFHOSTS)
           (AND [LISTP (SETQ PROTOCOLS (GETHOSTINFO HOST 'PROTOCOLS])
                 (NOT (MEMB 'LEAF PROTOCOLS])
                (RETURN \LEAF.NEVER.OPENED))
            (NOT (SETQ IFSPORT (BESTPUPADDRESS HOST PROMPTWINDOW)))
            (RETURN))
          (EQ (CDR IFSPORT)
              0)
          (SETQ IFSPORT (CONS (CAR IFSPORT)
                             \SOCKET.LEAF])
        (SETQ NAME/PASS (\INTERNAL/GETPASSWORD HOST))
        (RETURN (WITH.MONITOR \LEAFCONNECTIONLOCK
                  (PROG (CONN RESULT DATA OPUP)
                    [SETQ CONN (create SEQUIN
                                       SEQNAME _ HOST
                                       SEQFRNPORT _ IFSPORT
                                       SEQACKED _ (FUNCTION \LEAF.ACKED)
                                       SEQINPUT _ (FUNCTION \LEAF.HANDLE.INPUT)
                                       SEQBROKEN _ (FUNCTION \LEAF.FIX.BROKEN.SEQUIN)
                                       SEQABORTED _ (FUNCTION \LEAF.FIX.BROKEN.SEQUIN)
                                       SEQTIMEDOUT _ (FUNCTION \LEAF.TIMEDOUT)
                                       SEQTIMEDIN _ (FUNCTION \LEAF.TIMEDIN)
                                       SEQCLOSED _ (FUNCTION \LEAF.WHENCLOSED)
                                       SEQIDLEFN _ (FUNCTION \LEAF.IDLE)
                                       SEQIDLETIMEOUTCOMPUTER _ (FUNCTION \LEAF.IDLE?)
                                       SEQOPENERERRORHANDLER _ (FUNCTION \LEAF.OPENERERRORHANDLER)
                                       SEQDONEQ _ (NCREATE 'SYSQUEUE)
                                       LEAFCACHETIMER _ (\CREATECELL \FIXP)
                                       SEQIGNOREDUPPLICATES _ T
                                       LEAFOPENCLOSELOCK _ (CREATE.MONITORLOCK (CONCAT HOST "#LEAFOPEN"]

```

; NOTE: Implicit RESETLST

```

(INITSEQUIN CONN (PACK* HOST "#LEAF"))
(replace (SEQUIN LEAFCACHEHITS) of CONN with 0)
(replace (SEQUIN LEAFCACHEMISSES) of CONN with 0)
(RESETSAVE NIL (LIST [FUNCTION (LAMBDA (SEQUIN)
                      (AND RESETSTATE (\SEQUIN.CLOSE SEQUIN)
                                      CONN))
                      ]))
RETRY
  (PROGN (SETQ OPUP (ALLOCATE.PUP)) ; Build a LEAF RESET op
         (SETQ DATA (fetch PUPCONTENTS of OPUP))
         (\CLEARBYTES DATA 0 \LEN.RESETLEAF)
         (replace (LEAFDATA LEAFOPCODE) of DATA with \LEAFOP.RESET)
         (replace (LEAFDATA LEAFLNGTH) of DATA with \LEN.RESETLEAF)
         (\ADDLEAFSTRING OPUP (CAR NAME/PASS))
         (\ADDLEAFSTRING OPUP (CDR NAME/PASS)
          T)
         (replace PUPLNGTH of OPUP with (+ (fetch (LEAFDATA LEAFLNGTH) of DATA)
                                           \PUPOVLEN)))
         (replace EPUSERFIELD of OPUP with NIL)
         (replace (LEAFPACKET LEAFFLAGS) of OPUP with (LOGOR \LF.ALLOWERRORS \LF.WANTANSWER))
         (PUTSEQUIN CONN OPUP)
         (until (SELECTC (fetch (SEQUIN SEQSTATE) of CONN)
                   (\SS.OPENING ; still waiting for an answer
                    NIL)
                   (\SS.OPEN ; Connection has become open, or already was if this is a retry
                    (SETQ RESULT (fetch EPUSERFIELD of OPUP)))
                   (PROGN ; Some bad state
                    (SETQ RESULT (fetch EPUSERFIELD of OPUP))
                    T))
         (do (AWAIT.EVENT (fetch (SEQUIN SEQEVENT) of CONN)
                \ETHERTIMEOUT))
         (SELECTC RESULT
          ((LIST NIL \LEAF.BROKEN.STATUS)
           (RETURN NIL))
          (\LEAF.NEVER.OPENED
           (RETURN \LEAF.NEVER.OPENED))
          NIL)
         (COND
          ((EQ (fetch (LEAFDATA LEAFOPCODE) of (fetch PUPCONTENTS of RESULT))
              \LEAFOP.ERROR)
           (SELECTC (SETQ RESULT (PROG1 (fetch (LEAFERRORDATA LEAFERRORCODE)
                                             of (fetch PUPCONTENTS of RESULT))
                                       (RELEASE.PUP RESULT)))
                    (\PASSWORD.ERRORS ; Password error
                     (COND
                      ((SETQ NAME/PASS (\FIXPASSWORD RESULT CONN))
                       (GO RETRY))))
                    NIL)
           (\SEQUIN.CLOSE CONN)
           (RETURN NIL)))
         (RELEASE.PUP RESULT)
         (LET [(TIMEOUT (TIMES 2 (IQUOTIENT \LEAF.IDLETIMEOUT 5000))
              ;; Build a LEAF PARAMS op, making the connection timeout be twice the time that we would time it out
              ;; ourselves (so as to reduce the likelihood that the server would kill us without our consent).
              (SETQ OPUP (ALLOCATE.PUP))
              (SETQ DATA (fetch PUPCONTENTS of OPUP))
              (\CLEARBYTES DATA 0 \LEN.LEAFPARAMS)
              (replace (LEAFDATA LEAFOPCODE) of DATA with \LEAFOP.PARAMS)
              (replace (LEAFDATA LEAFLNGTH) of DATA with \LEN.LEAFPARAMS)
              (replace (LEAFPARAMSDATA LEAFPCONNTIMEOUT) of DATA with TIMEOUT)
              (replace (LEAFPARAMSDATA LEAFPLOCKTIMEOUT) of DATA with TIMEOUT)
              ; Make lock timeout the same, so we don't have silly lock broken
              ; stuff to worry about.
              (replace PUPLNGTH of OPUP with (+ \LEN.LEAFPARAMS \PUPOVLEN)))
              (replace EPUSERFIELD of OPUP with NIL)
              (replace (LEAFPACKET LEAFFLAGS) of OPUP with \LF.ALLOWERRORS)
              (PUTSEQUIN CONN OPUP)
              (RETURN CONN))]))

```

(\LEAF.BREAKCONNECTION

[LAMBDA (HOST DEVICE FAST)

; Edited 24-May-91 15:12 by jds

;;; Breaks connection to host, if there is one. Returns T if it broke something, NIL if there was nothing to break. If FAST is true, does not attempt to  
;;; cleanly close any files open on the host

```

(LET (CONNECTION FILES DEVINFO)
  (COND
   ((AND (type? PUPFILESERVER (SETQ DEVINFO (fetch DEVICEINFO of DEVICE)))
          (SETQ CONNECTION (fetch (PUPFILESERVER PFSLEAFSEQUIN) of DEVINFO)))
    [COND
     ((SETQ FILES (FDEVOP 'OPENP DEVICE NIL NIL DEVICE))
      (COND
       (FAST (for S in FILES do (FDEVOP 'UNREGISTERFILE DEVICE DEVICE S)))
       (T (MAPC FILES (FUNCTION CLOSEF)
                  (\CLOSELEAFCONNECTION CONNECTION DEVICE]))

```

(\CLOSELEAFCONNECTION

```
[LAMBDA (CONN DEVICE) ; Edited 24-May-91 14:53 by jds
  (PROG1 [COND
    ((CLOSESEQUIN CONN)
     (fetch (SEQUIN SEQNAME) of CONN))
    (T (LIST (fetch (SEQUIN SEQNAME) of CONN)
             'aborted]
      (replace (LEAFDEVICE PFSLEAFSEQUIN) of DEVICE with NIL)))]])
```

(\LEAF.EVENTFN

```
[LAMBDA (FDEV EVENT-TYPE) ; Edited 24-May-91 15:12 by jds
```

;;; Called before LOGOUT etc to clean up any leaf connections we have open

```
(PROG ((DEVINFO (fetch DEVICEINFO of FDEV))
      CONNECTION SOC)
  (SELECTQ EVENT-TYPE
    ((BEFORELOGOUT BEFOREMAKESYS BEFORESYSOUT)
     (COND
      ((SETQ CONNECTION (fetch (PUPFILESERVER PFSLEAFSEQUIN) of DEVINFO))
       (\FLUSH.OPEN.STREAMS FDEV)
       ;; Would like to have a monitor on this to prevent other processes from writing files now, but it can't be the main sequin
       ;; lock
       (\CLOSELEAFCONNECTION CONNECTION FDEV))))
     ((AFTERLOGOUT AFTERMAKESYS AFTERSYSOUT AFTERSAVEVM)
      (COND
       ((SETQ CONNECTION (fetch (PUPFILESERVER PFSLEAFSEQUIN) of DEVINFO))
        (\SEQUIN.FLUSH.CONNECTION CONNECTION \SS.ABORT)))
       (COND
        ((NOT (FDEVOP 'OPENP FDEV NIL NIL FDEV))
         ;; Association between hostname and host goes away over logout, so flush it. If there is a file open on it, however,
         ;; assume it's okay
         (\REMOVEDEVICE FDEV)))
        (COND
         ((SETQ SOC (fetch (PUPFILESERVER PFSLOOKUPFILESOCKET) of DEVINFO))
          (CLOSEPUPSOCKET SOC)
          (replace (PUPFILESERVER PFSLOOKUPFILESOCKET) of DEVINFO with NIL)))
         (replace (PUPFILESERVER PFSLOOKUPFILELOCK) of DEVINFO with NIL)
          ; revalidate open files
         (\PAGED.REVALIDATEFILELST FDEV))
        NIL]))
  NIL))
```

;; This generic fn ought to be on FILEIO

(DEFINEQ

(\BREAKCONNECTION

```
[LAMBDA (HOST FAST) ; Edited 23-Dec-87 12:29 by bvm:
```

;;; User entry. Breaks connection to host, if there is one, or all hosts if host = t. Returns name of any device that handled it. If FAST is true, may not attempt to cleanly close any files open on the host

```
(LET (DEVICE BREAKFN)
  (COND
   ((EQ HOST T)
    (for DEV in \FILEDEVICES when (AND (SETQ BREAKFN (fetch BREAKCONNECTION of DEV))
                                       (CL:FUNCALL BREAKFN (fetch DEVICENAME of DEV)
                                                           DEV FAST))
     collect (fetch DEVICENAME of DEV)))
   ((AND [OR (SETQ DEVICE (\GETDEVICEFROMNAME HOST T T))
             (AND (SETQ HOST (CANONICAL.HOSTNAME HOST))
                  (SETQ DEVICE (\GETDEVICEFROMNAME HOST T T))
                  (SETQ BREAKFN (fetch BREAKCONNECTION of DEVICE))
                  (CL:FUNCALL BREAKFN (fetch DEVICENAME of DEVICE)
                                      DEVICE FAST))
            (fetch DEVICENAME of DEVICE)]))
  )
```

;; Functions called when various SEQUIN events occur

(DEFINEQ

(\LEAF.ACKED

```
[LAMBDA (PUP SEQUIN) ; Edited 24-May-91 14:53 by jds
```

;; Called when a packet has been acked

```
(\ENQUEUE (fetch (SEQUIN SEQDONEQ) of SEQUIN)
  PUP)
(add (fetch (SEQUIN SEQINPUTQLENGTH) of SEQUIN)
```

1])

(\LEAF.FIX.BROKEN.SEQUIN

; Edited 24-May-91 15:08 by jds

[LAMBDA (SEQUIN PUP)

:: Called when BROKEN received. Try to open a new connection, and transfer everything over

```

(PROG ((STATE (fetch (SEQUIN SEQSTATE) of SEQUIN))
      (RETRANSQ (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN))
      (ACKEDQ (fetch (SEQUIN SEQDONEQ) of SEQUIN))
      (DEVICE (\GETDEVICEFROMNAME (fetch (SEQUIN SEQNAME) of SEQUIN)))
      UNANSWEREDPUPS AFFECTEDFILES NEWCONNECTION STRM)
      (\SEQUIN.FLUSH.RETRANSMIT SEQUIN)
      (COND
        (PUP ; Attempt to send PUP on a broken connection
          (GO GET.NEW.CONNECTION)))
      [COND
        ((SETQ UNANSWEREDPUPS (fetch SYSQUEUEHEAD of ACKEDQ))
          ; There were acked but not answered packets, so process them
          ; ahead of the unacked ones
          (replace EPLINK of (fetch SYSQUEUEHEAD of ACKEDQ) with (fetch SYSQUEUEHEAD of RETRANSQ))
          (replace SYSQUEUEHEAD of ACKEDQ with (replace SYSQUEUEHEAD of ACKEDQ with NIL)))
        (T (SETQ UNANSWEREDPUPS (fetch SYSQUEUEHEAD of RETRANSQ)
          [SELECTC STATE
            (\SS.OPENING ; Probably means we crashed on this local machine a while back using exactly the same socket number, so leaf
              ; thinks we're confused. This virtually never happens now that we choose Pup sockets more cleverly
                (COND
                  ((AND UNANSWEREDPUPS (NOT (fetch EPLINK of UNANSWEREDPUPS))
                    (EQ (fetch (LEAFDATA LEAFOPCODE) of (fetch PUPCONTENTS of UNANSWEREDPUPS))
                      \LEAFOP.RESET))
                    [replace (SEQUIN SEQSOCKET) of SEQUIN with (PROG1 (OPENPUPSOCKET)
                      ; Get a new socket and try again
                      (CLOSEPUPSOCKET (fetch (SEQUIN
                        SEQSOCKET
                        )
                          of SEQUIN)))]
                    (replace PUPSOURCECKET of UNANSWEREDPUPS with 0)
                    ; Let SENDPUP fill in the new socket
                    (RETURN (\SEQUIN.RETRANSMIT SEQUIN)))
                    (T (GO FAILURE))))
                  ((LIST \SS.OPEN \SS.CLOSING)
                    (COND
                      ((NULL UNANSWEREDPUPS)
                        ; No activity has gone unanswered here, so safe to just abort the
                        ; connection
                        (\SEQUIN.FLUSH.CONNECTION SEQUIN)
                        (RETURN T))))
                      (GO FAILURE))
                    ; This SEQUIN is bad, probably because of a file server crash (or we were idle a long time and it timed us out) so flush it and try to establish a
                    ; new one, retransmitting anything that wasn't yet answered
                    (replace SYSQUEUEHEAD of RETRANSQ with (replace SYSQUEUEHEAD of RETRANSQ with NIL))
                    ; Detach old queues of packets from dead connection
                    (printout PROMPTWINDOW "[Connection with " (fetch (SEQUIN SEQNAME) of SEQUIN)
                      " crashed; " "trying to establish new connection...")
                    GET.NEW.CONNECTION
                    (SETQ AFFECTEDFILES (for STREAM in (FDEVOP 'OPENP DEVICE NIL NIL DEVICE) collect STREAM
                      when (EQ (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
                        SEQUIN)))
                    RETRY.NEW.CONNECTION
                    [COND
                      ((SETQ NEWCONNECTION (\LEAF.RECONNECT DEVICE (AND (EQ (fetch (SEQUIN LEAFABORTSTATUS) of SEQUIN)
                        'ABORT)
                          (NOT (\CLOCKGREATERP (fetch (SEQUIN SEQTIMER)
                            of SEQUIN)
                              \LEAF.RECOVERY.TIMEOUT]
                          ; Succeeded in getting a new connection, so restore files
                          (\SEQUIN.FLUSH.CONNECTION SEQUIN)
                          [COND
                            ((AND \WINDOWWORLD (NOT (HASTTYWINDOWP)))
                              ; Assure that output from what follows has enough space to print. Note that this does not actually open the window (though it
                              ; may create it). Also, we don't care about restoration on exit, because this process is doomed anyway.
                              (WINDOWPROP T 'PAGEFULLFN (FUNCTION EXPANDING-PAGEFULLFN)
                              [COND
                                (PUP ; Attempt to send PUP on a broken connection
                                  (AND AFFECTEDFILES (\PAGED.REVALIDATEFILELST DEVICE))
                                  (RETURN (\LEAF.REPAIR.BROKEN.PUP SEQUIN PUP)))
                                  ((NOT (SETQ UNANSWEREDPUPS (\LEAF.USE.NEW.CONNECTION NEWCONNECTION UNANSWEREDPUPS
                                    AFFECTEDFILES)))
                                    (printout PROMPTWINDOW "done]" T)
                                    (RETURN T)
                                  [COND
                                    ((NULL (fetch (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN))
                                      (\SEQUIN.FLUSH.CONNECTION SEQUIN))
                                    ((forDuration 60000 do (COND
                                      ((EQ (fetch (SEQUIN LEAFABORTSTATUS) of SEQUIN)

```

```

        /ABORT)
        (\SEQUIN.FLUSH.CONNECTION SEQUIN)
        (RETURN T)))
        (AWAIT.EVENT (fetch (SEQUIN SEQEVENT) of SEQUIN)
         5000))
        (RETURN))
        (T (GO RETRY.NEW.CONNECTION)))
;; Either failed to make the new connection or something happened to the file
FAILURE
[ERROR "File server connection has been broken--cannot complete file operation(s). (RETURN) to try
again to get a new connection." (COND
        ((AND PUP (SETQ STRM (fetch EPUSERFIELD of PUP)))
         (.NAMEORSTREAM. STRM))
         (T (fetch (SEQUIN SEQNAME) of SEQUIN)]
(GO RETRY.NEW.CONNECTION])

```

**(\LEAF.REPAIR.BROKEN.PUP**

```

[LAMBDA (OLDSEQUIN PUP) ; Edited 2-Nov-92 03:37 by sybalsky:mv:envos
;; PUP is a pup that we were trying to send on a dead sequin. If we have since established the new connection, there is a new sequin in PUP's
;; stream, and we can patch the pup. Returns the new connection, or NIL if it can't
(PROG ((STREAM (fetch EPUSERFIELD of PUP))
      NEWCONNECTION DATA)
[COND
  ((OR (NULL STREAM)
       (NOT (type? STREAM STREAM))) ; Not much to go on
   )
  ((AND (SETQ NEWCONNECTION (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM))
        (NEQ NEWCONNECTION OLDSEQUIN)
        (SELECTC (fetch (LEAFDATA LEAFOPCODE) of (SETQ DATA (fetch PUPCONTENTS of PUP)))
                 ((LIST \LEAFOP.READ \LEAFOP.WRITE \LEAFOP.TRUNCATE \LEAFOP.DELETE \LEAFOP.CLOSE)
                  ; These operations all have their handle in the same place
                  (replace (LEAFDATA HANDLE) of DATA with (fetch (LEAFSTREAM LEAFHANDLE) of STREAM))
                  T)
                 NIL))
        (RETURN NEWCONNECTION))
  (T (ERROR "File server connection broken" (OR (fetch (STREAM FULLFILENAME) of STREAM)
                                                STREAM]
(replace (LEAFPACKET LEAFSTATUS) of PUP with \LEAF.BROKEN.STATUS)
(RETURN NIL])

```

**(\LEAF.USE.NEW.CONNECTION**

```

[LAMBDA (SEQUIN UNSENTPUPS AFFECTEDFILES) ; Edited 2-Nov-92 03:37 by sybalsky:mv:envos
(PROG (BUSYFILES OPCODE OLDSTREAM PUP DATA GOODPUPS BADPUPS RESENDPUPS)
      (while UNSENTPUPS do [SETQ PUP (COND
        ((LISTP UNSENTPUPS) ; We're given a list of packets, so hand them back one at a time.
         (POP UNSENTPUPS))
        (T ; Given a single packet, follow the normal queue line field.
         (PROG1 UNSENTPUPS
              (SETQ UNSENTPUPS (fetch EPLINK of UNSENTPUPS))))
(replace EPLINK of PUP with NIL)
(SELECTC [SETQ OPCODE (fetch (LEAFDATA LEAFOPCODE) of (SETQ DATA
                                                              (fetch PUPCONTENTS
                                                                of PUP]
         ((LIST \LEAFOP.READ \LEAFOP.WRITE \LEAFOP.TRUNCATE \LEAFOP.DELETE)
          ; These operations all have their handle in the same place
          (COND
            ((SETQ OLDSTREAM (fetch EPUSERFIELD of PUP))
             (pushnew AFFECTEDFILES OLDSTREAM)
             (pushnew BUSYFILES OLDSTREAM)
             (push GOODPUPS PUP))
            (T ; Shouldn't happen
             (push BADPUPS PUP))))
(\LEAFOP.CLOSE
 [COND
  ((SETQ OLDSTREAM (fetch EPUSERFIELD of PUP))
   (COND
    ((FMEMB OLDSTREAM BUSYFILES)
     ; There are other operations on this file, so include the close
     (push GOODPUPS PUP))
    ((DIRTYABLE OLDSTREAM)
     (push BUSYFILES OLDSTREAM))
    (T ; Closing a file open only for read; don't bother
     (SETQ AFFECTEDFILES (DREMOVE OLDSTREAM AFFECTEDFILES]))
(\LEAFOP.OPEN
  ;; just trying to open a file, so should work fine with the new connection; however, \LEAF.GETFILE
  ;; needs to know to use the new connection, so easier to just mark it broken here
  (replace (LEAFPACKET LEAFSTATUS) of PUP with \LEAF.BROKEN.STATUS))
(push BADPUPS PUP)))
(for STREAM in (UNION BUSYFILES AFFECTEDFILES) when (DIRTYABLE STREAM)
 do (printout T T "*****Warning: " (fetch (STREAM FULLFILENAME) of STREAM)
    " was open for write during a file server crash; data may be lost" T T))

```

```

(COND
  (AFFECTEDFILES (SETQ AFFECTEDFILES (\PAGED.REVALIDATEFILES AFFECTEDFILES))
    ; Reopen those files, make sure they still exist and haven't been
    ; modified
  ))
[for PUP in GOODPUPS do ; Do operation with new handle
  (COND
    ((FMEMB (SETQ OLDSTREAM (fetch EPUSERFIELD of PUP))
      AFFECTEDFILES)
      (replace (LEAFDATA HANDLE) of (fetch PUPCONTENTS of PUP)
        with (fetch (LEAFSTREAM LEAFHANDLE) of OLDSTREAM))
      (push RESENDPUPS PUP))
    (T (push BADPUPS PUP)
      [COND
        (RESENDPUPS (ADD.PROCESS (LIST '\LEAF.RESENDPUPS (KWOTE SEQUIN)
          (KWOTE RESENDPUPS)
        (RETURN BADPUPS]))

```

(\LEAF.RESENDPUPS

```

[LAMBDA (SEQUIN PUPS) ; (* bvm%: "17-APR-83 18:10")
  (while PUPS do (replace PUPSOURCESOCKET of (CAR PUPS) with 0)
    (PUTSEQUIN SEQUIN (pop PUPS))

```

(\LEAF.HANDLE.INPUT

```

[LAMBDA (PUP SEQUIN) ; Edited 24-May-91 15:08 by jds
  ; Called when a data sequin arrives

```

```

(PROG ((PUPDATA (fetch PUPCONTENTS of PUP))
  DONEPUP DONEPUPDATA ERROR OPCODE STREAM)
  ;; Under current scheme, where every requesting packet is responded to by exactly one packet, we 'know' that PUP matches up with the head of
  ;; SEQDONEQ. The error checking here is thus for protocol violation and is optional
  (SETQ DONEPUP (\DEQUEUE (fetch (SEQUIN SEQDONEQ) of SEQUIN))
  [COND
    ((NOT DONEPUP)
      (RETURN (SHOULDNT "Leaf lost a packet somewhere!")
    (add (fetch (SEQUIN SEQINPUTQLENGTH) of SEQUIN)
      -1)
    [COND
      ((EQ (fetch (LEAFDATA ANSWERBIT) of PUPDATA)
        0)
        (HELP "Leaf Protocol violation--will terminate connection" (fetch (SEQUIN SEQNAME) of SEQUIN))
        (RETURN (RELEASE.PUP PUP)
      (COND
        ((EQ (SETQ OPCODE (fetch (LEAFDATA LEAFOPCODE) of PUPDATA))
          \LEAFOP.ERROR)
          (SETQ OPCODE (fetch (LEAFERRORDATA LEAFERROROPCODE) of PUPDATA))
          (SETQ ERROR T)))
        (COND
          ((AND (NEQ (fetch (LEAFDATA LEAFOPCODE) of (SETQ DONEPUPDATA (fetch PUPCONTENTS of DONEPUP)))
            OPCODE)
            LEAFDEBUGFLG) ; Protocol violation, but the buggy Vax server does this
            (HELP "Answer does not match head of done queue" PUP))
          ([AND ERROR (NOT (fetch (LEAFPACKET LEAFALLOWERRORS) of DONEPUP))
            (NOT (AND (EQ OPCODE \LEAFOP.CLOSE)
              (EQ (fetch (LEAFERRORDATA LEAFERRORCODE) of PUPDATA)
                \IFSERROR.BAD.HANDLE]
            ;; Last clause says that if we were closing the file and got a bad handle error, to ignore it -- this typically happens if two files try to
            ;; close the same file simultaneously
            (replace (LEAFPACKET LEAFSTATUS) of PUP with (fetch (LEAFERRORDATA LEAFERRORCODE) of DONEPUPDATA))
            (SETQ STREAM (fetch EPUSERFIELD of DONEPUP))
            (COND
              ((type? STREAM STREAM)
                (add (fetch (LEAFSTREAM LEAFERRORCNT) of STREAM)
                  1)))
            (replace EPUSERFIELD of DONEPUP with PUP)
            (ADD.PROCESS (LIST (FUNCTION \LEAF.ERROR)
              PUP
              (KWOTE STREAM)
              (SEQUIN DONEPUP)))
            ((fetch (LEAFPACKET LEAFANSWERWANTED) of DONEPUP)
              ;; Match the request with its response; requestor will watch this slot. Eventually change this to a NOTIFY
            (replace EPUSERFIELD of DONEPUP with PUP))
            (T (RELEASE.PUP PUP)
              (RELEASE.PUP DONEPUP])

```

(\LEAF.OPENERROHANDLER

```

[LAMBDA (SEQUIN PUP) ; Edited 24-May-91 14:54 by jds
  (SELECTC (fetch ERRORPUPCODE of PUP)
    (\PUPE.NOSOCKET
      (printout PROMPTWINDOW T "[No Leaf Server on " (fetch (SEQUIN SEQNAME) of SEQUIN))
      (COND

```

```

(\FTPAVAILABLE (printout PROMPTWINDOW "; trying FTP..."))
(printout PROMPTWINDOW "]")
\SS.NOSOCKET)
(\PUPE.NOROUTE
(printout PROMPTWINDOW T "[No route to " (fetch (SEQUIN SEQNAME) of SEQUIN)
"]")
T)
NIL])

```

**(\LEAF.TIMEDIN**

```

[LAMBDA (SEQUIN) ; Edited 24-May-91 14:54 by jds
(COND
((fetch (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN)
(CLOSEW (fetch (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN))
(replace (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN with NIL)
(replace (SEQUIN LEAFABORTSTATUS) of SEQUIN with NIL)))
(replace (SEQUIN LEAFTIMEOUTSTATUS) of SEQUIN with NIL])

```

**(\LEAF.TIMEDOUT**

```

[LAMBDA (SEQUIN CNT) ; Edited 24-May-91 14:54 by jds
; The SEQTIMEDOUT fn for LEAF

```

```

(COND
(EQ (fetch (SEQUIN LEAFABORTSTATUS) of SEQUIN)
'ABORT)
(\SEQUIN.CONTROL SEQUIN \SEQUIN.BROKEN)
(\SEQUIN.FLUSH.CONNECTION SEQUIN))
(>= CNT \MAXLEAFTRIES)
(PROG ((TRIES (fetch (SEQUIN LEAFTIMEOUTSTATUS) of SEQUIN))
(STATE (fetch (SEQUIN SEQSTATE) of SEQUIN))
PUP)
(if (NULL TRIES)
then ; First time partner is slow
(SELECTC STATE
(\SS.OPENING ; can't open connection
(\LEAF.NOT.RESPONDING SEQUIN :OPEN PROMPTWINDOW)
(\SEQUIN.FLUSH.CONNECTION SEQUIN))
(\SS.OPEN (if (SETQ PUP (\LEAF.STREAM.IN.QUEUE SEQUIN))
then ; Something is going on worth mentioning. If the only thing in the
; queue is us trying to close the cache, say, we keep quiet
(\LEAF.NOT.RESPONDING SEQUIN PUP PROMPTWINDOW)
(COND
(PUPTRACEFLG (\LEAF.NOT.RESPONDING SEQUIN PUP PUPTRACEFILE)
(TERPRI PUPTRACEFILE)))
(replace (SEQUIN LEAFTIMEOUTSTATUS) of SEQUIN with CNT)))
(\SS.CLOSING [COND
(NULL (SETQ PUP (\LEAF.STREAM.IN.QUEUE SEQUIN T)))
; Safe to abort connection, since no information left to be acked
(COND
(PUPTRACEFLG (printout PUPTRACEFILE T "[File server connection to
" (fetch (SEQUIN SEQNAME) of SEQUIN)
" aborted"])))
(RETURN (\SEQUIN.FLUSH.CONNECTION SEQUIN)
(\LEAF.NOT.RESPONDING SEQUIN PUP PROMPTWINDOW))
NIL)
elseif (EQ CNT (+ TRIES \MAXLEAFTRIES))
then ; Enough, already, better let us get out
(\LEAF.TIMEDOUT.EXCESSIVE SEQUIN CNT])

```

**(\LEAF.NOT.RESPONDING**

```

[LAMBDA (SEQUIN REASON OUTSTREAM) ; Edited 24-May-91 14:54 by jds
; Alearts user that connection not responding. REASON is from some unacked packet in the queue, or :OPEN if trying to open the connection.
(printout OUTSTREAM T "[" (fetch (SEQUIN SEQNAME) of SEQUIN)
" not responding")
(SELECTQ REASON
(T ; T means those silly nonsense name directory requests
)
(:OPEN (printout OUTSTREAM " to Leaf connection attempt")
(printout OUTSTREAM " for " (.NAMEORSTREAM. REASON)))
(printout OUTSTREAM "]" ]))

```

**(\LEAF.TIMEDOUT.EXCESSIVE**

```

[LAMBDA (SEQUIN CNT) ; Edited 2-Nov-92 03:37 by sybalsky:mv:envos
(AND (WINDOWWORLDP)
(PROG ([W (CREATEW (MAKEWITHINREGION LEAFABORTREGION)
(CONCAT "Leaf Abort window for " (fetch (SEQUIN SEQNAME) of SEQUIN)
(PUP (fetch SYSQUEUEHEAD of (fetch (SEQUIN SEQDONEQ) of SEQUIN)))
(FIRSTTIME T)
READFILES WRITEFILES X DATA PAGE FULLNAME)
(replace (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN with W)
(printout W (fetch (SEQUIN SEQNAME) of SEQUIN)
" is not responding." T)
(PROG NIL

```





```

((AND (SETQ X (fetch EPUSERFIELD of PUP))
 (OR (NOT (type? STREAM X))
      (fetch (STREAM FULLFILENAME) of X))
 (if (NOT IMPORTANT)
      then (if (EQ X T)
                then (SETQ DEFAULT T)
                     NIL)
      else T)
 (elseif (type? STREAM X)
          then (SELECTC (fetch (LEAFDATA LEAFOPCODE) of (fetch PUPCONTENTS of PUP))
                      ((LIST \LEAFOP.WRITE \LEAFOP.TRUNCATE)
                       ; Always important
                       T)
                      (\LEAFOP.CLOSE
                       ; Closing an output file?
                       (DIRTYABLE X))
                      NIL)))
 (RETURN X)))
(SETQ PUP (fetch EPLINK of PUP))
(GO LP])

```

(\LEAF.IDLE

[LAMBDA (SEQUIN) ; Edited 24-May-91 15:08 by jds

;; Called after a suitable timeout with no activity on connection

```

(COND
 [(fetch (SEQUIN LEAFCACHEDFILE) of SEQUIN)
  (ADD.PROCESS (LIST (FUNCTION \LEAF.MAYBE.FLUSH.CACHE)
                    (KWOTE SEQUIN)
                    ((for STREAM in (fetch (FDEV OPENFILELST) of (\GETDEVICEFROMNAME (fetch (SEQUIN SEQNAME) of SEQUIN)))
                      thereis (EQ (fetch (LEAFSTREAM LEAFCONNECTION) of STREAM)
                                  SEQUIN))
                  ; Keep activity on this connection
                      (\SEQUIN.CONTROL SEQUIN \SEQUIN.NOOP))
                    (T (replace (SEQUIN LEAFCLOSING) of SEQUIN with T)
                       (\SEQUIN.CLOSE SEQUIN)))]])

```

(\LEAF.MAYBE.FLUSH.CACHE

[LAMBDA (SEQUIN) ; Edited 24-May-91 14:54 by jds

;; Called when leaf connection has been idle a while and there is a file in the cache. Only flush it if we can get the lock; else try again later. This  
;; keeps this process from hanging (and identical ones accumulating) in the case where the connection is wedged.

```

(if (OBTAIN.MONITORLOCK (fetch (SEQUIN LEAFOPENCLOSELOCK) of SEQUIN)
    T T)
    then (\LEAF.FLUSH.CACHE SEQUIN])

```

(\LEAF.WHENCLOSED

[LAMBDA (SEQUIN FINALSTATE REASON) ; Edited 24-May-91 15:12 by jds

```

(PROG ((CODE (COND
          ((EQ REASON \SS.NOSOCKET)
           \LEAF.NEVER.OPENED)
          (T \LEAF.BROKEN.STATUS)))
      PUP DEV)
 (replace (SEQUIN LEAFCACHEDFILE) of SEQUIN with NIL) ; Break this potential circular link
 (COND
  ((fetch (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN)
   (CLOSEW (fetch (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN))
   (replace (SEQUIN LEAFABORTBUTTONWINDOW) of SEQUIN with NIL)))
 (while (SETQ PUP (\DEQUEUE (fetch (SEQUIN SEQDONEQ) of SEQUIN))) do (replace (LEAFPACKET LEAFSTATUS)
                                                                              of PUP with CODE))
 (while (SETQ PUP (\DEQUEUE (fetch (SEQUIN SEQRETRANSMITQ) of SEQUIN)))
   do (replace (LEAFPACKET LEAFSTATUS) of PUP with CODE))
 (replace (SEQUIN SEQINPUTLENGTH) of SEQUIN with 0)
 (AND (SETQ DEV (\GETDEVICEFROMNAME (fetch (SEQUIN SEQNAME) of SEQUIN)
                                         T T))
      (EQ (fetch (PUPFILESERVER PFSLEAFSEQUIN) of (SETQ DEV (fetch DEVICEINFO of DEV)))
          SEQUIN)
      (replace (PUPFILESERVER PFSLEAFSEQUIN) of DEV with NIL]))

```

(\LEAF.IDLE?

[LAMBDA (SEQUIN) ; Edited 24-May-91 14:54 by jds

;; Tells SEQUIN process how long to block when it otherwise has nothing to do, i.e. no packets remain unacked

```

(COND
 ((NEQ (fetch (SEQUIN SEQINPUTLENGTH) of SEQUIN)
        0)
  NIL)
 ((fetch (SEQUIN LEAFCACHEDFILE) of SEQUIN)
  \LEAF.CACHETIMEOUT)
 (T
  \LEAF.IDLETIMEOUT])

```

)

(ADDTOVAR NETWORKOSTYPES )

:: Miscellaneous and error handling

(DEFINEQ

(\ADDLEAFSTRING

```
[LAMBDA (PUP STRING DECODE) ; Edited 24-May-91 14:58 by jds
  (PROG ((PUPBASE (fetch PUPCONTENTS of PUP))
    LEAFLEN STRLEN STRBASE STROFF PUPSTRBASE NEWLENGTH)
    (SETQ LEAFLEN (CEIL (fetch (LEAFDATA LEAFLENGTH) of PUPBASE)
      BYTESPERWORD)) ; Round Length up to next word--strings must be word-aligned
  (COND
    ((NULL STRING)
      (SETQ STRLEN 0))
    ((LITATOM STRING)
      (SETQ STRBASE (fetch (LITATOM PNAMEBASE) of STRING))
      (SETQ STROFF 1)
      (SETQ STRLEN (fetch (LITATOM PNAMELENGTH) of STRING)))
    (T (OR (STRINGP STRING)
      (SETQ STRING (MKSTRING STRING)))
      (SETQ STRBASE (fetch (STRINGP BASE) of STRING))
      (SETQ STROFF (fetch (STRINGP OFFST) of STRING))
      (SETQ STRLEN (fetch (STRINGP LENGTH) of STRING))
    (COND
      ((IGREATERP (SETQ NEWLENGTH (IPLUS LEAFLEN STRLEN BYTESPERWORD))
        \MAX.PUPLength)
        (ERROR "PUP OVERFLOW" PUP)))
    (\PUTBASE (SETQ PUPSTRBASE (\ADDBASE PUPBASE (FOLDLO LEAFLEN BYTESPERWORD)))
      0 STRLEN)
    (SETQ PUPSTRBASE (\ADDBASE PUPSTRBASE 1))
    (COND
      ((EQ STRLEN 0))
      [DECODE (for I from 0 to (SUB1 STRLEN) do (\PUTBASEBYTE PUPSTRBASE I (\DECRYPT.PWD.CHAR
        (\GETBASEBYTE STRBASE
          (IPLUS I STROFF))
        (T (\MOVEBYTES STRBASE STROFF PUPSTRBASE 0 STRLEN)))
      (replace (LEAFDATA LEAFLENGTH) of PUPBASE with NEWLENGTH])
```

(\FIXPASSWORD

```
[LAMBDA (ERRCODE CONNECTION DIRECTORY) ; Edited 24-May-91 14:54 by jds
  :: Called when a username or password error occurs. ERRCODE is the IFS errorcode (name or password error). Attempts to get new name
  :: and/or password for use on CONNECTION. If DIRECTORY is specified, it is a connect error to that directory
  (\INTERNAL/GETPASSWORD (fetch (SEQUIN SEQNAME) of CONNECTION)
    (NEQ ERRCODE \IFSERROR.PROTECTION)
    DIRECTORY
    (SELECTC ERRCODE
      (\IFSERROR.PASSWORD
        "Incorrect password")
      ((LIST \IFSERROR.USERNAME \IFSERROR.NEED.USERNAME)
        "Invalid username")
      (\IFSERROR.CONNECTPASSWORD
        "Incorrect connect password")
      (\IFSERROR.CONNECTNAME
        "Invalid connect name")
      (\IFSERROR.PROTECTION
        "Protection violation")
      (\IFSERROR.NO.LOGIN
        "Can't login as files-only directory")
      "Unknown error"])
```

(\GETLEAFSTRING

```
[LAMBDA (ADDR) (* bvm%: "30-MAR-83 17:39")
  :: Retrieves the IFS string starting at ADDR. IFS string has length in its first word
  (PROG ((LEN (\GETBASE ADDR 0)))
    (RETURN (AND (IGREATERP LEN 0)
      (\GETBASESTRING ADDR 2 LEN])
```

(\IFSERRORSTRING

```
[LAMBDA (CODE FILENAME CONNECTION) ; Edited 24-May-91 14:54 by jds
  :: Returns the error string associated with IFS error CODE. FILENAME is the name of the file that caused the error (used for recursion break);
  :: CONNECTION is the leaf connection on which the error occurred
  (COND
    ((NOT (AND FILENAME (STRING.EQUAL FILENAME \IFSERRORFILENAME))))
    (LET* ([ERR-MSG-STREAM (CAR (NLSETQ (OPENSTREAM (SETQ \IFSERRORFILENAME
      (PACK* '{ (COND
        (CONNECTION (fetch (SEQUIN SEQNAME)
          of CONNECTION))
        (T \CONNECTED.HOST))
      "}<SYSTEM>IFS.ERRORS")
      ' INPUT]
```

```
(ERR-FILE-NAME (FULLNAME ERR-MSG-STREAM))
(EOL (FCHARACTER (CHARCODE EOL)))
(START NIL)
(LEN NIL)
(RESET NIL))
```

;; This is a text file containing entries that look like '\$\$<error code> <error message>'. Entries can extend over one line. Entries are  
 ;; sorted by error code, but I don't make use of that knowledge in the brute force procedure below

```
(COND
  (ERR-MSG-STREAM (SETQ \IFSERRORFILENAME ERR-FILE-NAME)
                  ; In case an error happens while scanning file, update this var to
                  ; correct value
    (PROG1 (COND
      ((SETQ START (FFILEPOS (CONCAT EOL "$$" CODE " " )
                              ERR-MSG-STREAM 0 NIL NIL T))
        (SETQ LEN (IDIFFERENCE (OR (FFILEPOS (CONCAT EOL "$$")
                                             ERR-MSG-STREAM START)
                                   (GETEOFPTR ERR-MSG-STREAM))
                              START))
          ; Length of entry
        (SETQ RESULT (ALLOCSTRING LEN))
        (SETFILEPTR ERR-MSG-STREAM START)
        (for I from 1 to LEN do (RPLCHARCODE RESULT I (\BIN ERR-MSG-STREAM)))
        (CLOSEF ERR-MSG-STREAM))
```

(\LEAF.ERROR

; Edited 2-Nov-92 03:37 by sybalsky:mv:envos

```
[LAMBDA (PUP FILENAME CONNECTION SENTPUP)
  (PROG ((DATA (fetch PUPCONTENTS of PUP))
        (ERRCODE MSG)
        (RETURN (SELECTC (SETQ ERRCODE (fetch (LEAFERRORDATA LEAFERRORCODE) of DATA))
          (\IFSERROR.FILE.NOT.FOUND
            (LISPERROR "FILE NOT FOUND" FILENAME))
          (\IFSERROR.MALFORMED
            (LISPERROR "BAD FILE NAME" FILENAME))
          (\IFSERROR.ALLOCATION
            (LISPERROR "FILE SYSTEM RESOURCES EXCEEDED" FILENAME))
          (\IFSERROR.BAD.HANDLE
            (ERROR "Leaf Error: Bad Handle.
              This shouldn't happen: Lisp and the server have different ideas about which
              file they are talking about. All operations to this file are now suspended.
              See a wizard if possible." (fetch (STREAM FULLFILENAME) of FILENAME)))
          (PROGN [SETQ MSG (SELECTC ERRCODE
            (\IFSERROR.BUSY
              "File busy")
            (\IFS.ERROR.BROKEN.LEAF
              "Leaf Broken--a file you had open was accessed by another user
              while it was idle."
              (CONCAT "Leaf error: "
                (OR [AND (IGREATERP (fetch PUPLength of PUP)
                                   \SHORT.ERROR.PUPLen)
                    (\GETLEAFSTRING (LOCf (fetch (LEAFERRORDATA
                                                  LEAFERRORMSG)
                                                  of DATA)
                    (\IFSERRORSTRING ERRCODE FILENAME CONNECTION)
                    ERRCODE])
                (COND
                  ((EQ (fetch (LEAFERRORDATA LEAFERROROPCODE) of DATA)
                      \LEAFOP.OPEN)
                    (printout PROMPTWINDOW T MSG T)
                    (LISPERROR "FILE WON'T OPEN" FILENAME))
                  (T (ERROR MSG FILENAME]))
```

(\LEAF.DIRECTORYNAMEONLY

(\* bvm%: "19-NOV-81 11:34")

```
[LAMBDA (FILENAME)
  (PROG ((DIR (FILENAMEFIELD FILENAME 'DIRECTORY))
        N)
    (RETURN (COND
      ((SETQ N (STRPOS '> DIR))
        (SUBATOM DIR 1 (SUB1 N)))
      (T DIR]))
```

(GETHOSTINFO

; Edited 10-Oct-90 17:39 by gadener

```
[LAMBDA (HOST ATTRIBUTE)
  (SETQ HOST (MKATOM (U-CASE HOST)))
  (PROG (NSFLG (INFO (ASSOC HOST NETWORKOSTYPES))
        VAL)
    (COND
      (INFO ; already know about this host
        )
      [ (SETQ NSFLG (STRPOS '%: HOST))
        (SETQ INFO '(NIL . NS)) ; default NS information
        [ (AND (BOUNDP \IPFLG) ; Check for IP info
              \IPFLG)
          (SETQ HOST (\DOMAIN.NAME.QUALIFY.FULLY HOST))
```

```

    (SETQ INFO (CONS NIL (fetch (HOSTS.TXT.ENTRY HTE.OS.TYPE) of (GETHASH HOST \IP.HOSTNAMES]
    [(AND (NEQ HOST (SETQ HOST (CANONICAL.HOSTNAME HOST)))
    ; Check for NS and PUP info

    (SETQ INFO (ASSOC HOST NETWORKKOSTYPES]
    (DEFAULT.OSTYPE (SETQ INFO (CONS NIL DEFAULT.OSTYPE)))
    (T (RETURN)))
    (RETURN (OR (SELECTQ ATTRIBUTE
    ((OS OSTYPE) ; get OS type
    (COND
    ((LISTP (CDR INFO))
    (LISTGET (CDR INFO)
    'OSTYPE))
    (T (CDR INFO))))
    (LOGINFO [COND
    ((SETQ VAL (ASSOC HOST NETWORKLOGINFO))
    (CDR VAL))
    (T (CDR (ASSOC (COND
    ((LISTP (CDR INFO))
    (LISTGET (CDR INFO)
    'OSTYPE))
    (T (CDR INFO)))
    NETWORKLOGINFO]))
    (PROTOCOLS (COND
    ((LITATOM (CDR INFO))
    (SELECTQ (CDR INFO)
    (IFS '(LEAF PUPFTP CHAT LOOKUPFILE)
    NIL))))
    (AND (LISTP (CDR INFO))
    (LISTGET (CDR INFO)
    ATTRIBUTE]))

```

(GETOSTYPE

```

[LAMBDA (HOST) ; (* bvm%: "31-OCT-83 17:08")
  (GETHOSTINFO HOST 'OSTYPE)]

```

(EXPANDING-PAGEFULLFN

```

[LAMBDA (W) ; Edited 14-Apr-87 22:25 by bvm:
  ;; Hack for getting a window large enough to hold everything you want to display without having to make it big enough in the first place. This
  ;; function is intended to be the PAGEFULLFN on the window that is your process's ttydisplaystream. As soon as the window fills up, it grows the
  ;; window on the bottom to show more. The number of lines it expands by is given by the window's EXPANDING-INCREMENT property, defaults
  ;; to 4.
  (LET ((OLDREGION (WINDOWREGION W))
        [INCREMENT (TIMES (OR (WINDOWPROP W 'EXPANDING-INCREMENT)
                              4)
                              (- (DSPLINEFEED NIL W)
                                  (CURRENTHEIGHT \#DISPLAYLINES))
        [SHAPEW W (create REGION using OLDREGION HEIGHT _ (+ INCREMENT (fetch (REGION HEIGHT) of OLDREGION))
                  BOTTOM _ (IMAX 0 (- (fetch (REGION BOTTOM) of OLDREGION)
                                      INCREMENT])
  ;; The SHAPEW resets height parameters as if window cleared. We want display to believe that the pagefullfn never happened, so that we
  ;; can expand again the next time we hit bottom.
  (SETQ \CURRENTDISPLAYLINE CURRENTHEIGHT))
)

```

```

(RPAQQ DEFAULT.OSTYPE IFS)

```

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```

```

(GLOBALVARS DEFAULT.OSTYPE)
)

```

;; LookUpFile stuff

```

(DEFINEQ

```

(IFS.LOOKUPFILE

```

[LAMBDA (NAME RECOG ATTRIBUTE DEVINFO) ; Edited 24-May-91 15:12 by jds

```

;;; Attempt to use the LookupFile protocol to get full filename

```

(PROG ((RESULT '?)
  (HOSTNAME (fetch (PUPFILESERVER PFSNAME) of DEVINFO))
  (OSTYPE (fetch (PUPFILESERVER PFSOSTYPE) of DEVINFO))
  REMOTENAME SEMI NAME/PASS START DOT ROOTNAME INFO IPUP OPUP PUPSOC DIREND LOCK)
(COND
  ([OR (NEQ (NTHCHARCODE NAME 1)
            (CHARCODE {}))
    (NOT (SETQ START (STRPOS ' } NAME 2]
    (RETURN)))
(COND
  ((NOT (SETQ LOCK (fetch (PUPFILESERVER PFSLOOKUPFILELOCK) of DEVINFO)))

```

```

; First time to do this
(replace (PUPFILESERVER PFSLOOKUPFILESOCKET) of DEVINFO with (SETQ PUPSOC (OPENPUPSOCET)))
(replace (PUPFILESERVER PFSLOOKUPFILELOCK) of DEVINFO with (SETQ LOCK (CREATE.MONITORLOCK
"LookUpFile")))
(replace (PUPFILESERVER PFSLOOKUPFAILCNT) of DEVINFO with 0))
(NOT (SETQ PUPSOC (fetch (PUPFILESERVER PFSLOOKUPFILESOCKET) of DEVINFO)))
(RETURN RESULT)))
[SETQ ROOTNAME (SUBSTRING NAME (ADD1 START)
(COND
([SETQ SEMI (OR (STRPOS ';' NAME (ADD1 START))
(STRPOS '! NAME (ADD1 START))
(PROG1 (SUB1 SEMI)
(COND
(EQ SEMI (NCHARS NAME))
; Not really a version there
(SETQ SEMI NIL)))]))
(while (SETQ DOT (STRPOS '> ROOTNAME DIREND)) do (SETQ DIREND (ADD1 DOT)))
[COND
(NOT DIREND)
(SETQ DIREND (IMINUS (NCHARS ROOTNAME)))
(SETQ ROOTNAME (CONCAT '< (CAR (\INTERNAL/GETPASSWORD HOSTNAME))
'> ROOTNAME])
(COND
[(SETQ DOT (STRPOS '%. ROOTNAME DIREND))
; Name ends in dot, but is only "extensionless" if the dot isn't
; quoted
(SETQ DOT (AND (EQ DOT (NCHARS ROOTNAME))
(NEQ (NTHCHARCODE ROOTNAME (SUB1 DOT))
(CHARCODE "' '))
(T (SETQ ROOTNAME (CONCAT ROOTNAME '%.))
(SETQ DOT T)))]
; DOT now T if filename is extensionless. ROOTNAME is
; everything but the version
[SETQ REMOTENAME (COND
[(EQ (SETQ OSTYPE (GETHOSTINFO HOSTNAME 'OSTYPE))
'TENEX)
; Our filenames are already Tenex style
(COND
((OR SEMI (NEQ RECOG 'OLDEST))
ROOTNAME)
(T (CONCAT ROOTNAME "-2")
[SEMI
; Use ! for version delimiter
(CONCAT (COND
(DOT (SUBSTRING ROOTNAME 1 -2))
(T ROOTNAME))
(COND
((EQ OSTYPE 'TOPS20)
'%.))
(T '!))
(SUBSTRING NAME (ADD1 SEMI])
((EQ OSTYPE 'TOPS20)
(COND
((EQ RECOG 'OLDEST)
(CONCAT ROOTNAME "-2"))
(T ROOTNAME)))
(T (SETQ REMOTENAME (COND
(DOT (SUBSTRING ROOTNAME 1 -2))
(T ROOTNAME)))
(COND
((EQ RECOG 'OLDEST)
(CONCAT REMOTENAME "!L"))
(T REMOTENAME]
(WITH.MONITOR LOCK
(SETUPPUP (SETQ OPUP (ALLOCATE.PUP))
HOSTNAME \SOCKET.LOOKUPFILE \PT.LOOKUPFILE NIL PUPSOC)
(\PUTPUPSTRING OPUP (if (STRPOS "' " REMOTENAME)
then (\LEAF.STRIP.QUOTES REMOTENAME)
else REMOTENAME))
[to \MAXETHERTRIES when (SETQ IPUP (EXCHANGEPUPS PUPSOC OPUP NIL T))
do (SELECTC (fetch PUPTYPE of IPUP)
(\PT.LOOKUPFILEREPLY
[RETURN (SETQ RESULT (SELECTQ ATTRIBUTE
((NAME NIL)
(SETQ REMOTENAME (CONCAT '{ HOSTNAME ' } ROOTNAME
';
(fetch (LOOKUPFILEDATA
LOOKUPVERSION)
of IPUP)))
(if *UPPER-CASE-FILE-NAMES*
then (MKATOM (U-CASE REMOTENAME))
else REMOTENAME))
(CREATIONDATE (GDATE (ALTO.TO.LISP.DATE
(fetch (LOOKUPFILEDATA
LOOKUPCREATIONDATE)
of IPUP))))
(ICREATIONDATE
(ALTO.TO.LISP.DATE (fetch (LOOKUPFILEDATA
LOOKUPCREATIONDATE)
of IPUP)))
(LENGTH (fetch (LOOKUPFILEDATA LOOKUPLength) of IPUP))

```

```

(\ILLEGAL.ARG ATTRIBUTE]]
(\PT.LOOKUPFILEERROR ; No such file
(RETURN (SETQ RESULT NIL)))
(\PT.ERROR (COND
  ((EQ (fetch ERRORPUPCODE of IPUP)
        \PUPE.NOSOCKET) ; No such socket
    (AND PUPTRACEFLG (PRINTERORPUP IPUP PUPTRACEFILE))
    (replace (PUPFILESERVER PFSLOOKUPFILESOCKET) of DEVINFO with NIL)
    (CLOSEPUPSOCKET PUPSOC)
    (RETURN))))
NIL)
(RELEASE.PUP IPUP)
finally (SETQ IPUP)
(COND
  (PUPTRACEFLG "LookupFile timed out" T))
(COND
  ((AND (fetch (PUPFILESERVER PFSLOOKUPFAILCNT) of DEVINFO)
        (> (add (fetch (PUPFILESERVER PFSLOOKUPFAILCNT) of DEVINFO)
                1)
            4))
    (replace (PUPFILESERVER PFSLOOKUPFILESOCKET) of DEVINFO with NIL)
    (CLOSEPUPSOCKET PUPSOC])
  (AND IPUP (RELEASE.PUP IPUP))
  (COND
    ((NEQ RESULT '?')
     (replace (PUPFILESERVER PFSLOOKUPFAILCNT) of DEVINFO with NIL))))
(RETURN RESULT])
)

```

(DECLARE%: EVAL@COMPILE DONTCOPY

```

(RPAQQ LOOKUPFILECOMS ((CONSTANTS \PT.LOOKUPFILE \PT.LOOKUPFILEREPLY \PT.LOOKUPFILEERROR \SOCKET.LOOKUPFILE)
  (RECORDS LOOKUPFILEDATA)
  (GLOBALVARS \LOOKUPFILE.HOSTINFO))
)

```

(DECLARE%: EVAL@COMPILE

(RPAQQ \PT.LOOKUPFILE 128)

(RPAQQ \PT.LOOKUPFILEREPLY 129)

(RPAQQ \PT.LOOKUPFILEERROR 130)

(RPAQQ \SOCKET.LOOKUPFILE 49)

```

(CONSTANTS \PT.LOOKUPFILE \PT.LOOKUPFILEREPLY \PT.LOOKUPFILEERROR \SOCKET.LOOKUPFILE)
)

```

(DECLARE%: EVAL@COMPILE

```

[ACCESSFNS LOOKUPFILEDATA ((LOOKUPFILEBASE (fetch PUPCONTENTS of DATUM))
  (BLOCKRECORD LOOKUPFILEBASE ((LOOKUPVERSION WORD)
    (LOOKUPCREATIONDATE FIXP)
    (LOOKUPLLENGTH FIXP]
)
)

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \LOOKUPFILE.HOSTINFO)

)

(DEFINEQ

(\LEAFINIT

```

[LAMBDA NIL (* bvm%: "12-SEP-83 15:39")
  (SETQ \LEAFCONNECTIONLOCK (CREATE.MONITORLOCK 'LEAF))
  (\DEFINEDEVICE NIL (create FDEV
    DEVICENAME _ 'LEAF
    RESETABLE _ T
    RANDOMACCESSP _ T
    PAGEMAPPED _ T
    HOSTNAMEP _ (FUNCTION \LEAF.DEVICEP)
    EVENTFN _ (FUNCTION NIL)
    DELETEFILE _ (FUNCTION \ILLEGAL.DEVICEOP)
    GETFILEINFO _ (FUNCTION \ILLEGAL.DEVICEOP)
    OPENFILE _ (FUNCTION \ILLEGAL.DEVICEOP)
    SETFILEINFO _ (FUNCTION \ILLEGAL.DEVICEOP)
    GETFILENAME _ (FUNCTION \ILLEGAL.DEVICEOP)
    GENERATEFILES _ (FUNCTION \ILLEGAL.DEVICEOP)
    DIRECTORYNAMEP _ (FUNCTION \ILLEGAL.DEVICEOP)
    RENAMEFILE _ (FUNCTION \ILLEGAL.DEVICEOP])
)
)

```

(DECLARE%: DONTVAL@LOAD

(\LEAFINIT)  
)

(DEFINEQ

(PRINTLEAF

[LAMBDA (PUP)

; Edited 24-May-91 14:59 by jds

::: Prints a LEAF pup. Called from PRINTPUP

```
(PROG ((LENGTH (IDIFFERENCE (fetch PUPLENGTH of PUP)
                             \PUPOVLEN))
      DATA OP START HI LO MACRO NBYTES)
  (COND
    ((EQ (fetch (SEQUINPACKET SEQCONTROL) of PUP)
         \SEQUIN.DATA)
      (printout NIL "SequinData"))
    (T (printout NIL "SequinOp = ")
       (PRINTCONSTANT (fetch (SEQUINPACKET SEQCONTROL) of PUP)
                       SEQUINOPS NIL "\SEQUIN.)))
    (printout NIL ", alloc = " .P2 (fetch (SEQUINPACKET ALLOCATE) of PUP)
      ", recv = " .P2 (fetch (SEQUINPACKET RECEIVESEQ) of PUP)
      ", send = " .P2 (fetch (SEQUINPACKET SENDSEQ) of PUP)
      T)
  [COND
    ((IGREATERP LENGTH 0)
      (SETQ DATA (fetch PUPCONTENTS of PUP))
      (printout NIL "Leaf")
      (COND
        ((SETQ OP (SELECTC (fetch (LEAFDATA LEAFOPCODE) of DATA)
                           (\LEAFOP.OPEN "Open")
                           (\LEAFOP.CLOSE "Close")
                           (\LEAFOP.READ "Read")
                           (\LEAFOP.WRITE "Write")
                           (\LEAFOP.ERROR "Error")
                           NIL))
          (printout NIL OP))
        (T (printout NIL "Op = ")
           (PRINTCONSTANT (fetch (LEAFDATA LEAFOPCODE) of DATA)
                           LEAFOPCODES NIL "\LEAFOP.)))
      (COND
        ((EQ (fetch (LEAFDATA ANSWERBIT) of DATA)
             1)
          (printout NIL " (ans)"))
        (COND
          ((AND (EQ (fetch (LEAFDATA OPCODE) of DATA)
                     \LEAFOP.WRITE)
                (EQ (fetch (LEAFDATA EOFBIT) of DATA)
                     1))
            (printout NIL " (eof)"))
          (COND
            ((NEQ (fetch (LEAFDATA LEAFLENGTH) of DATA)
                  LENGTH)
              (printout NIL ", length = " .P2 (fetch (LEAFDATA LEAFLENGTH) of DATA)
                " [but Pup Length = header + " .P2 LENGTH "!]"))
            (printout NIL ", Handle = " .P2 (fetch (LEAFDATA HANDLE) of DATA))
            (COND
              ([AND (IGREATERP LENGTH (SETQ START 4))
                    (SETQ MACRO
                        (SELECTC (fetch (LEAFDATA LEAFOPCODE) of DATA)
                                (\LEAFOP.OPEN [COND
                                  ((EQ (fetch (LEAFDATA ANSWERBIT) of DATA)
                                       0)
                                    '("Mode: " WORDS 6 " Login: " CHARS IFSSTRING ; BYTES IFSSTRING
                                      " Connect: " CHARS IFSSTRING ; BYTES IFSSTRING " File: "
                                      CHARS IFSSTRING))
                                  (T '("FileLength = " INTEGER 8 |...|])
                                (\LEAFOP.RESET
                                  '("Login: " CHARS IFSSTRING BYTES))
                                (LIST \LEAFOP.READ \LEAFOP.WRITE)
                                  (SETQ HI (SIGNED (fetch (LEAFDATA JUSTHIADDR) of DATA)
                                                  11))
                                  (SETQ LO (fetch (LEAFDATA LOADDR) of DATA))
                                  (SETQ NBYTES (fetch (LEAFDATA DATALENGTH) of DATA))
                                  [COND
                                    [(AND (EVENP NBYTES BYTESPERPAGE)
                                           (IGEQ HI 0))
                                      [printout NIL ", Page " .P2 (SETQ LO (IPLUS (FOLDLO LO BYTESPERPAGE)
                                                                                   (LLSH HI 7))
                                      (COND
                                        ((IGREATERP NBYTES BYTESPERPAGE)
                                          (printout NIL " thru " .P2 (IPLUS LO (FOLDLO NBYTES BYTESPERPAGE)
                                                                                   -1]
                                        (printout NIL " thru " .P2 (IPLUS LO (FOLDLO NBYTES BYTESPERPAGE)
                                                                                   -1]
                                  ]
                                ]
              ]
            ]
          ]
        ]
      ]
    ]
  ]
)
```

(T (printout NIL T .P2 NBYTES " bytes from " .P2 (\MAKENUMBER (UNSIGNED HI BITSPERWORD)

LO])

[COND ((SELECTC (fetch (LEAFDATA LEAFOPCODE) of DATA) (\LEAFOP.WRITE (EQ (fetch (LEAFDATA ANSWERBIT) of DATA) 0)) (IGREATERP LENGTH 10)) (SETQ START 10) '("Data: " CHARS 20 |...|])) (\LEAFOP.ERROR '("Error op: " WORDS 6 "Error handle: " 8 IFSSTRING)) ' (BYTES]

(TERPRI) (PRINTPACKETDATA DATA START MACRO LENGTH)) (T (TERPRI]

(TERPRI))

PUP])

)

(ADDTOVAR PUPPRINTMACROS (176 . PRINTLEAF))

(RPAQ? LEAFDEBUGFLG )

(RPAQ? LEAFABORTREGION ' (417 616 399 192))

(RPAQ? \MAXLEAFTRIES 4)

(RPAQ? NOFILEPROPERROR )

(RPAQ? DEFAULTFILETYPE ' TEXT)

(RPAQ? \SOCKET.LEAF 35)

(RPAQ? \SEQUIN.TIMEOUTMAX 10000)

(RPAQ? \LEAF.IDLETIMEOUT 1800000)

(RPAQ? \LEAF.CACHETIMEOUT 90000)

(RPAQ? \LEAF.MAXCACHE 10)

(RPAQ? \LEAF.RECOVERY.TIMEOUT 600000)

(RPAQ? \LEAF.MAXLOOKAHEAD 4)

(RPAQ? \FTPAVAILABLE )

(RPAQ? UNIXFTPFLG )

(RPAQ? NONLEAFHOSTS )

(RPAQ? \*UPPER-CASE-FILE-NAMES\* T)

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQ? LEAFCOMPILETIMECOMS

((RECORDS LEAFDATA LEAFERRORDATA LEAFPARAMSDATA LEAFPACKET LEAFINFOBLOCK LEAFSTREAM LEAFDEVICE PUPFILESERVER)

(MACROS .NAMEORSTREAM. .PAGE.IS.AFTER.EOF. INCLEAFSTAT)

(CONSTANTS \* LEAFOPCODES)

(CONSTANTS \* IFSERRORS)

(CONSTANTS (\PT.LEAF 176)

(\PT.ERROR 4)

(\LEAFOP.ANSWERBIT 1024)

(\LEAF.READBIT 32768)

(\LEAF.WRITEBIT 16384)

(\LEAF.EXTENDBIT 8192)

(\LEAF.MULTIBIT 4096)

(\LEAF.CREATEBIT 2048)

(\LEAF.DEFAULT.LOWEST 128)

(\LEAF.DEFAULT.HIGHEST 256)

(\LEAF.DEFAULT.NEXT 384)

(\LEAF.EXPLICIT.ANY 1536)

(\LEAF.EXPLICIT.OLD 512)

(\LEAF.EXPLICIT.NEXT.OR.OLD 1024)

(\LEN.RESETLEAF 4)

(\LEN.LEAFPARAMS 8)

(\LEN.NOOPREQUEST 2)

(\LEN.OPENREQUEST 6)

(\LEN.FILEREREQUEST 10)

(\LEN.CLOSEREQUEST 4)

(\LEN.READANSWER 10)

(\OPCODE.SHIFT 11)



```
(\LEN.CLOSEREQUEST 4)
(\MAXLEN.FILENAME 100)
(\OFFSET.FILENAME (TIMES 2 256))
(\BYTES.PER.TRIDENT.PAGE 2048)
(\LEN.DATE 4)
(\LEAFMODE.DONTEXTEND 2)
(\LEN.FILETYPE&SIZE 4)
(\OFFSET.FILETYPE 680)
(\OFFSET.BACKUPDATE 676)
(\OFFSET.AUTHOR 636)
(\LEN.AUTHOR 40)
(\SHORT.ERROR.PUPLLEN 30)
(\LEAF.GOODSTATUS 65534)
(\LF.ALLOWERRORS 2)
(\LF.WANTANSWER 1)
(\LEAF.BROKEN.STATUS 65529)
(\LEAF.NEVER.OPENED 65531)
(CONSTANTS (\FT.TEXT 1)
(\FT.BINARY 2)
(\FT.UNKNOWN 0))
(LOCALVARS . T)
(GLOBALVARS \LEAFDEVICE \SOCKET.LEAF LEAFDEBUGFLG PUPTRACEFLG NOFILEPROPERROR NETWORKOSTYPES LEAFOPCODES
SEQUINOPS DEFAULTFILETYPE \LEAF.IDLETIMEOUT \LEAF.CACHETIMEOUT \LEAF.MAXLOOKAHEAD \LEAF.MAXCACHE
\LEAFCONNECTIONLOCK \FTPAVAILABLE UNIXFTPFLG \SEQUIN.TIMEOUTMAX LEAFABORTREGION \MAXLEAFTRIES
\LEAF.RECOVERY.TIMEOUT NONLEAFHOSTS \FTPFDEV))
```

(DECLARE%: EVAL@COMPILE

```
(BLOCKRECORD LEAFDATA ((OPWORD WORD)
(HANDLE WORD)
(FILEADDRESS FIXP)
(DATALLENGTH WORD)
(LEAFFIRSTDATAWORD WORD)) ; Format of typical file operation request.
(BLOCKRECORD LEAFDATA ((LEAFOPCODE BITS 5)
(ANSWERBIT BITS 1)
(LEAFLENGTH BITS 10)
(NIL WORD)
(READWRITEMODE BITS 2)
(EOFBIT BITS 1)
(NIL BITS 2)
(JUSTHIADDR BITS 11)
(LOADDR WORD)) ; Details of the file address format
(SYNONYM LEAFOPCODE (OPCODE)))
(BLOCKRECORD LEAFDATA ((NIL 2 WORD)
(SIGNEXTEND BITS 5)
(NIL BITS 27)) ; more details
)
(BLOCKRECORD LEAFDATA ((NIL 2 WORD)
(OPENMODE WORD)) ; format of OPEN file request
)
(BLOCKRECORD LEAFDATA ((NIL 5 WORD)
(LEAFFILETYPE WORD)
(LEAFBYTESIZE WORD)) ; For accessing the file's TYPE
)
(BLOCKRECORD LEAFDATA ((NIL 5 WORD)
(LEAFFILEDATE FIXP)) ; Format of SETFILEINFO of CREATIONDATE request
))
(BLOCKRECORD LEAFERRORDATA ((NIL WORD)
(LEAFERRORCODE WORD) ; Error subcode in ERROR leafop
(LEAFERROROPCODE BITS 5) ; The OPCode in the Leaf packet provoking the error
(NIL BITS 11)
(LEAFERRORHANDLE WORD) ; The handle in the provoking op
(LEAFERRORMSG WORD) ; Actually IFSSTRING starting here
))
(BLOCKRECORD LEAFFPARAMSDATA ((NIL WORD)
(LEAFPMAXDATALLENGTH WORD)
(LEAFPLOCKTIMEOUT WORD) ; File Lock timeout, in units of 5 seconds
(LEAFPCONNTIMEOUT WORD) ; Overall connection timeout, same units
)
)
[ACCESSFNS LEAFPACKET ((LEAFSTATUS (fetch EPUSERFIELD of DATUM)
(replace EPUSERFIELD of DATUM with NEWVALUE))
(LEAFFLAGS (fetch EPFLAGS of DATUM)
(replace EPFLAGS of DATUM with NEWVALUE))
(LEAFANSWERWANTED (NEQ (LOGAND (fetch EPFLAGS of DATUM)
\LF.WANTANSWER)
0))
(LEAFALLOWERRORS (NEQ (LOGAND (fetch EPFLAGS of DATUM)
\LF.ALLOWERRORS)
```

0]

```
(BLOCKRECORD LEAFINFOBLOCK ((LFCREATIONDATE FIXP
(LFWRITE DATE FIXP)
(LFREADDATE FIXP))
; just like leader page
(BLOCKRECORD LEAFINFOBLOCK ((HICREATE WORD)
(LOCREATE WORD)
(HIWRITE WORD)
(LOWRITE WORD)
(HIREAD WORD)
(LOREAD WORD))
; for VALIDATION use
```

(CREATE (\ALLOCBLOCK 3))

```
[ACCESSFNS LEAFSTREAM ((LEAFCONNECTION (fetch F1 of DATUM)
(replace F1 of DATUM with NEWVALUE))
(LEAFHANDLE (fetch F2 of DATUM)
(replace F2 of DATUM with NEWVALUE))
(LEAFPAGECACHE (fetch F3 of DATUM)
(replace F3 of DATUM with NEWVALUE))
(LEAFINFO (fetch F4 of DATUM)
(replace F4 of DATUM with NEWVALUE))
(LEAFREALLYOPEN (fetch F5 of DATUM)
(replace F5 of DATUM with NEWVALUE))
(LEAFCACHECNT (fetch FW6 of DATUM)
(replace FW6 of DATUM with NEWVALUE))
(LEAFERRORCNT (fetch FW7 of DATUM)
(replace FW7 of DATUM with NEWVALUE])
```

```
[ACCESSFNS LEAFDEVICE ((PUPFILESERVER (fetch DEVICEINFO of DATUM)
(replace DEVICEINFO of DATUM with NEWVALUE])
```

(DATATYPE PUPFILESERVER (;; Info common to various pup protocols used on a file server, independent of whether a connection is now open

- (NIL BYTE)
- (PFSNAME POINTER)
- (PFSADDRESS POINTER) ; Pup address
- (PFSOSTYPE POINTER)
- (PFSLEAFFLG POINTER) ; Indicates something about whether LEAF is available
- (PFSLEAFSEQUIN POINTER) ; Pointer to SEQUIN for open leaf connection
- (PFSLEAFTIMER POINTER) ; Timeout for handling dead servers
- (PFSLOOKUPFILESOCKET POINTER) ; The Pup socket for LookupFile requests
- (PFSLOOKUPFILELOCK POINTER) ; Lock to secure it
- (PFSLOOKUPFALLCNT POINTER) ; Counter used until we know the service exists
- (PFSKNOWNDIRS POINTER) ; List of directories known to exist on this host (for
- ; DIRECTORYNAMEP)
- (NIL POINTER))

)

```
(/DECLAREDATATYPE 'PUPFILESERVER ' (BYTE POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
POINTER POINTER)
```

;; ---field descriptor list elided by lister---

' 24)

(DECLARE%: EVAL@COMPILE

```
(PUTPROPS .NAMEORSTREAM. MACRO (OPENLAMBDA (FILENAME)
(COND
((type? STREAM FILENAME)
(fetch FULLFILENAME of FILENAME))
(T FILENAME))))
```

```
(PUTPROPS .PAGE.IS.AFTER.EOF. MACRO [OPENLAMBDA (STREAM PAGE#)
(AND (IGEQ PAGE# (fetch EPAGE of STREAM))
(OR (NOT (IEQP (fetch EPAGE of STREAM)
PAGE#))
(EQ (fetch EOFFSET of STREAM)
0]))
```

```
(PUTPROPS INCLEAFSTAT MACRO ((X)
(change X (IPLUS16 DATUM 1))))
```

)

(RPAQQ LEAFOPCODES

- (\LEAFOP.ERROR 0)
- (\LEAFOP.OPEN 1)
- (\LEAFOP.CLOSE 2)
- (\LEAFOP.DELETE 3)
- (\LEAFOP.LENGTH 4)
- (\LEAFOP.TRUNCATE 5)
- (\LEAFOP.READ 6)
- (\LEAFOP.WRITE 7)
- (\LEAFOP.RESET 8)
- (\LEAFOP.NOOP 9)
- (\LEAFOP.TELNET 10)

(\LEAFOP.PARAMS 11))

(DECLARE%: EVAL@COMPILE

(RPAQQ \LEAFOP.ERROR 0)

(RPAQQ \LEAFOP.OPEN 1)

(RPAQQ \LEAFOP.CLOSE 2)

(RPAQQ \LEAFOP.DELETE 3)

(RPAQQ \LEAFOP.LENGTH 4)

(RPAQQ \LEAFOP.TRUNCATE 5)

(RPAQQ \LEAFOP.READ 6)

(RPAQQ \LEAFOP.WRITE 7)

(RPAQQ \LEAFOP.RESET 8)

(RPAQQ \LEAFOP.NOOB 9)

(RPAQQ \LEAFOP.TELNET 10)

(RPAQQ \LEAFOP.PARAMS 11)

(CONSTANTS (\LEAFOP.ERROR 0)

(\LEAFOP.OPEN 1)

(\LEAFOP.CLOSE 2)

(\LEAFOP.DELETE 3)

(\LEAFOP.LENGTH 4)

(\LEAFOP.TRUNCATE 5)

(\LEAFOP.READ 6)

(\LEAFOP.WRITE 7)

(\LEAFOP.RESET 8)

(\LEAFOP.NOOB 9)

(\LEAFOP.TELNET 10)

(\LEAFOP.PARAMS 11))

(RPAQQ **IFSERRORS**

((\IFSERROR.BAD.CHARACTER 202)

(\IFSERROR.MALFORMED '(201 202))

(\IFSERROR.FILE.NOT.FOUND 207)

(\IFSERROR.PROTECTION 208)

(\IFSERROR.BUSY 209)

(\IFSERROR.INVALID.DIRECTORY 210)

(\IFSERROR.ALLOCATION 211)

(\IFSERROR.USERNAME 216)

(\IFSERROR.PASSWORD 217)

(\IFSERROR.NO.LOGIN 218)

(\PASSWORD.ERRORS '(216 217 218 223))

(\IFSERROR.CONNECTNAME 219)

(\IFSERROR.CONNECTPASSWORD 220)

(\CONNECT.PASSWORD.ERRORS '(219 220))

(\IFSERROR.NEED.USERNAME 223)

(\IFS.ERROR.BROKEN.LEAF 1001)

(\IFSERROR.BAD.HANDLE 1011))

(DECLARE%: EVAL@COMPILE

(RPAQQ \IFSERROR.BAD.CHARACTER 202)

(RPAQQ \IFSERROR.MALFORMED (201 202))

(RPAQQ \IFSERROR.FILE.NOT.FOUND 207)

(RPAQQ \IFSERROR.PROTECTION 208)

(RPAQQ \IFSERROR.BUSY 209)

(RPAQQ \IFSERROR.INVALID.DIRECTORY 210)

(RPAQQ \IFSERROR.ALLOCATION 211)

(RPAQQ \IFSERROR.USERNAME 216)

(RPAQQ \IFSERROR.PASSWORD 217)

(RPAQQ \IFSERROR.NO.LOGIN 218)

(RPAQQ \PASSWORD.ERRORS (216 217 218 223))

(RPAQQ \IFSERROR.CONNECTNAME 219)

```

(RPAQQ \IFSERROR.CONNECTPASSWORD 220)
(RPAQQ \CONNECT.PASSWORD.ERRORS (219 220))
(RPAQQ \IFSERROR.NEED.USERNAME 223)
(RPAQQ \IFS.ERROR.BROKEN.LEAF 1001)
(RPAQQ \IFSERROR.BAD.HANDLE 1011)
(CONSTANTS (\IFSERROR.BAD.CHARACTER 202)
  (\IFSERROR.MALFORMED ' (201 202))
  (\IFSERROR.FILE.NOT.FOUND 207)
  (\IFSERROR.PROTECTION 208)
  (\IFSERROR.BUSY 209)
  (\IFSERROR.INVALID.DIRECTORY 210)
  (\IFSERROR.ALLOCATION 211)
  (\IFSERROR.USERNAME 216)
  (\IFSERROR.PASSWORD 217)
  (\IFSERROR.NO.LOGIN 218)
  (\PASSWORD.ERRORS ' (216 217 218 223))
  (\IFSERROR.CONNECTNAME 219)
  (\IFSERROR.CONNECTPASSWORD 220)
  (\CONNECT.PASSWORD.ERRORS ' (219 220))
  (\IFSERROR.NEED.USERNAME 223)
  (\IFS.ERROR.BROKEN.LEAF 1001)
  (\IFSERROR.BAD.HANDLE 1011))
)
(DECLARE%: EVAL@COMPILE
(RPAQQ \PT.LEAF 176)
(RPAQQ \PT.ERROR 4)
(RPAQQ \LEAFOP.ANSWERBIT 1024)
(RPAQQ \LEAF.READBIT 32768)
(RPAQQ \LEAF.WRITEBIT 16384)
(RPAQQ \LEAF.EXTENDBIT 8192)
(RPAQQ \LEAF.MULTIBIT 4096)
(RPAQQ \LEAF.CREATEBIT 2048)
(RPAQQ \LEAF.DEFAULT.LOWEST 128)
(RPAQQ \LEAF.DEFAULT.HIGHEST 256)
(RPAQQ \LEAF.DEFAULT.NEXT 384)
(RPAQQ \LEAF.EXPLICIT.ANY 1536)
(RPAQQ \LEAF.EXPLICIT.OLD 512)
(RPAQQ \LEAF.EXPLICIT.NEXT.OR.OLD 1024)
(RPAQQ \LEN.RESETLEAF 4)
(RPAQQ \LEN.LEAFPARAMS 8)
(RPAQQ \LEN.NOOPREQUEST 2)
(RPAQQ \LEN.OPENREQUEST 6)
(RPAQQ \LEN.FILEREQUEST 10)
(RPAQQ \LEN.CLOSEREQUEST 4)
(RPAQQ \LEN.READANSWER 10)
(RPAQQ \OPCODE.SHIFT 11)
(RPAQQ \LEN.CLOSEREQUEST 4)
(RPAQQ \MAXLEN.FILENAME 100)
(RPAQQ \OFFSET.FILENAME (TIMES 2 256))
(RPAQQ \BYTES.PER.TRIDENT.PAGE 2048)
(RPAQQ \LEN.DATE 4)
(RPAQQ \LEAFMODE.DONTEXTEND 2)

```

```

{MEDLEY}<sources>LEAF.;1

(RPAQQ \LEN.FILETYPE&SIZE 4)

(RPAQQ \OFFSET.FILETYPE 680)

(RPAQQ \OFFSET.BACKUPDATE 676)

(RPAQQ \OFFSET.AUTHOR 636)

(RPAQQ \LEN.AUTHOR 40)

(RPAQQ \SHORT.ERROR.PUPLN 30)

(RPAQQ \LEAF.GOODSTATUS 65534)

(RPAQQ \LF.ALLOWERRORS 2)

(RPAQQ \LF.WANTANSWER 1)

(RPAQQ \LEAF.BROKEN.STATUS 65529)

(RPAQQ \LEAF.NEVER.OPENED 65531)

(CONSTANTS (\PT.LEAF 176)
  (\PT.ERROR 4)
  (\LEAFOP.ANSWERBIT 1024)
  (\LEAF.READBIT 32768)
  (\LEAF.WRITEBIT 16384)
  (\LEAF.EXTENDBIT 8192)
  (\LEAF.MULTIBIT 4096)
  (\LEAF.CREATEBIT 2048)
  (\LEAF.DEFAULT.LOWEST 128)
  (\LEAF.DEFAULT.HIGHEST 256)
  (\LEAF.DEFAULT.NEXT 384)
  (\LEAF.EXPLICIT.ANY 1536)
  (\LEAF.EXPLICIT.OLD 512)
  (\LEAF.EXPLICIT.NEXT.OR.OLD 1024)
  (\LEN.RESETLEAF 4)
  (\LEN.LEAFPARAMS 8)
  (\LEN.NOOPREQUEST 2)
  (\LEN.OPENREQUEST 6)
  (\LEN.FILEREQUEST 10)
  (\LEN.CLOSEREQUEST 4)
  (\LEN.READANSWER 10)
  (\OPCODE.SHIFT 11)
  (\LEN.CLOSEREQUEST 4)
  (\MAXLEN.FILENAME 100)
  (\OFFSET.FILENAME (TIMES 2 256))
  (\BYTES.PER.TRIDENT.PAGE 2048)
  (\LEN.DATE 4)
  (\LEAFMODE.DONTEXTEND 2)
  (\LEN.FILETYPE&SIZE 4)
  (\OFFSET.FILETYPE 680)
  (\OFFSET.BACKUPDATE 676)
  (\OFFSET.AUTHOR 636)
  (\LEN.AUTHOR 40)
  (\SHORT.ERROR.PUPLN 30)
  (\LEAF.GOODSTATUS 65534)
  (\LF.ALLOWERRORS 2)
  (\LF.WANTANSWER 1)
  (\LEAF.BROKEN.STATUS 65529)
  (\LEAF.NEVER.OPENED 65531))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \FT.TEXT 1)

(RPAQQ \FT.BINARY 2)

(RPAQQ \FT.UNKNOWN 0)

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

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \LEAFDEVICE \SOCKET.LEAF LEAFDEBUGFLG PUPTRACEFLG NOFILEPROPERROR NETWORKOSTYPES LEAFOPCODES
  SEQUINOPS DEFAULTFILETYPE \LEAF.IDLETIMEOUT \LEAF.CACHETIMEOUT \LEAF.MAXLOOKAHEAD \LEAF.MAXCACHE
  \LEAFCONNECTIONLOCK \FTPAVAILABLE UNIXFTPFLG \SEQUIN.TIMEOUTMAX LEAFABORTREGION \MAXLEAFTRIES
  \LEAF.RECOVERY.TIMEOUT NONLEAFHOSTS \FTPFDEV)

```

)  
)

(/DECLAREDATATYPE 'PUPFILESERVER ' (BYTE POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER  
POINTER POINTER)

;; ---field descriptor list elided by lister---

' 24)

(ADDTOTVAR **SYSTEMRECLST**

(DATATYPE PUPFILESERVER ((NIL BYTE)  
(PFSNAME POINTER)  
(PFSADDRESS POINTER)  
(PFSOSTYPE POINTER)  
(PFSLEAFFLG POINTER)  
(PFSLEAFSEQUIN POINTER)  
(PFSLEAFTIMER POINTER)  
(PFSLOOKUPFILESOCKET POINTER)  
(PFSLOOKUPFILELOCK POINTER)  
(PFSLOOKUPFAILCNT POINTER)  
(PFSKNOWNDIRS POINTER)  
(NIL POINTER))))

### FUNCTION INDEX

BREAKCONNECTION	27	\LEAF.FIX.BROKEN.SEQUIN	28	\LEAF.SETVALIDATION	24
CLEAR.LEAF.CACHE	22	\LEAF.FLUSH.CACHE	22	\LEAF.STREAM.IN.QUEUE	32
CLOSESEQUIN	5	\LEAF.FORCEOUTPUT	22	\LEAF.STRIP.QUOTES	17
EXPANDING-PAGEFULLFN	36	\LEAF.GENERATEFILES	14	\LEAF.TIMEDIN	31
GETHOSTINFO	35	\LEAF.GETFILE	14	\LEAF.TIMEDOUT	31
GETOSTYPE	36	\LEAF.GETFILEDATES	17	\LEAF.TIMEDOUT.EXCESSIVE	31
GETSEQUIN	6	\LEAF.GETFILEINFO	17	\LEAF.TRUNCATEFILE	24
INITSEQUIN	6	\LEAF.GETFILEINFO.OPEN	17	\LEAF.USE.NEW.CONNECTION	29
LEAF.ASSURE.FINISHED	22	\LEAF.GETFILENAME	18	\LEAF.WHENCLOSED	33
PRINTLEAF	39	\LEAF.HANDLE.INPUT	30	\LEAF.WRITEPAGES	24
PUTSEQUIN	6	\LEAF.IDLE	33	\LEAFINIT	38
\ADDLEAFSTRING	34	\LEAF.IDLE?	33	\OPENLEAFCONNECTION	25
\CLOSELEAFCONNECTION	27	\LEAF.LOOKUPCACHE	21	\PARSE.REMOTE.FILENAME	16
\FIXPASSWORD	34	\LEAF.MAYBE.FLUSH.CACHE	33	\SENDLEAF	25
\GETLEAFSTRING	34	\LEAF.NOT.RESPONDING	31	\SEQUIN.CLEANUP	9
\IFS.LOOKUPFILE	36	\LEAF.OPENERHANDLER	30	\SEQUIN.CLOSE	8
\IFSERRORSTRING	34	\LEAF.OPENFILE	18	\SEQUIN.COMPARE	9
\LEAF.ABORT.FROMMENU	32	\LEAF.READFILENAME	19	\SEQUIN.CONTROL	6
\LEAF.ACKED	27	\LEAF.READFILEPROP	20	\SEQUIN.FLUSH.CONNECTION	8
\LEAF.ADD.QUOTES	20	\LEAF.READPAGES	20	\SEQUIN.FLUSH.RETRANSMIT	9
\LEAF.BREAKCONNECTION	26	\LEAF.RECONNECT	13	\SEQUIN.HANDLE.ACK	10
\LEAF.CLOSEFILE	11	\LEAF.RENAMEFILE	22	\SEQUIN.HANDLE.INPUT	9
\LEAF.CREATIONDATE	23	\LEAF.REOPENFILE	22	\SEQUIN.OUT.OF.THE.BLUE	10
\LEAF.DELETEFILE	12	\LEAF.REPAIR.BROKEN.PUP	29	\SEQUIN.PROCESS	7
\LEAF.DEVICEP	12	\LEAF.REQUESTPAGE	21	\SEQUIN.PUT	7
\LEAF.DIRECTORYNAMEONLY	35	\LEAF.RESENDPUPS	30	\SEQUIN.RETRANSMIT	11
\LEAF.DIRECTORYNAMEP	14	\LEAF.SETCREATIONDATE	23	\SEQUIN.RETRANSMITNEXT	11
\LEAF.ERROR	35	\LEAF.SETFILEINFO	23		
\LEAF.EVENTFN	27	\LEAF.SETFILETYPE	23		

### CONSTANT INDEX

\BYTES.PER.TRIDENT.PAGE	45	\LEAFOP.ANSWERBIT	45	\PT.LOOKUPFILEERROR	38
\CONNECT.PASSWORD.ERRORS	44	\LEAFOP.CLOSE	43	\PT.LOOKUPFILEREPLY	38
\FT.BINARY	45	\LEAFOP.DELETE	43	\PT.SEQUIN	4
\FT.TEXT	45	\LEAFOP.ERROR	43	\SC.AHEAD	4
\FT.UNKNOWN	45	\LEAFOP.LENGTH	43	\SC.DUPLICATE	4
\IFS.ERROR.BROKEN.LEAF	44	\LEAFOP.NOOP	43	\SC.EQUAL	4
\IFSERROR.ALLOCATION	44	\LEAFOP.OPEN	43	\SC.OUTOFRANGE	4
\IFSERROR.BAD.CHARACTER	44	\LEAFOP.PARAMS	43	\SC.PREVIOUS	4
\IFSERROR.BAD.HANDLE	44	\LEAFOP.READ	43	\SEQUIN.ACK	4
\IFSERROR.BUSY	44	\LEAFOP.RESET	43	\SEQUIN.BREAK	4
\IFSERROR.CONNECTNAME	44	\LEAFOP.TELNET	43	\SEQUIN.BROKEN	4
\IFSERROR.CONNECTPASSWORD	44	\LEAFOP.TRUNCATE	43	\SEQUIN.DALLYING	4
\IFSERROR.FILE.NOT.FOUND	44	\LEAFOP.WRITE	43	\SEQUIN.DATA	4
\IFSERROR.INVALID.DIRECTORY	44	\LEN.AUTHOR	45	\SEQUIN.DEFAULT.ALLOCATION	4
\IFSERROR.MALFORMED	44	\LEN.CLOSEREQUEST	45	\SEQUIN.DEFAULT.RETRANSMITMAX	4
\IFSERROR.NEED.USERNAME	44	\LEN.DATE	45	\SEQUIN.DESTROY	4
\IFSERROR.NO.LOGIN	44	\LEN.FILEREQUEST	45	\SEQUIN.NOOP	4
\IFSERROR.PASSWORD	44	\LEN.FILETYPE&SIZE	45	\SEQUIN.OBSOLETE.CLOSE	4
\IFSERROR.PROTECTION	44	\LEN.LEAFPARAMS	45	\SEQUIN.OPEN	4
\IFSERROR.USERNAME	44	\LEN.NOOPREQUEST	45	\SEQUIN.QUIT	4
\LEAF.BROKEN.STATUS	45	\LEN.OPENREQUEST	45	\SEQUIN.RESTART	4
\LEAF.CREATEBIT	45	\LEN.READANSWER	45	\SHORT.ERROR.PUPLFN	45
\LEAF.DEFAULT.HIGHEST	45	\LEN.RESETLEAF	45	\SOCKET.LOOKUPFILE	38
\LEAF.DEFAULT.LOWEST	45	\LF.ALLOWERRORS	45	\SS.ABORT	4
\LEAF.DEFAULT.NEXT	45	\LF.WANTANSWER	45	\SS.CLOSED	4
\LEAF.EXPLICIT.ANY	45	\MAXLEN.FILENAME	45	\SS.CLOSING	4
\LEAF.EXPLICIT.NEXT.OR.OLD	45	\OFFSET.AUTHOR	45	\SS.DALLYING	4
\LEAF.EXPLICIT.OLD	45	\OFFSET.BACKUPDATE	45	\SS.DESTROYED	4
\LEAF.EXTENDBIT	45	\OFFSET.FILENAME	45	\SS.NOSOCKET	4
\LEAF.GOODSTATUS	45	\OFFSET.FILETYPE	45	\SS.OPEN	4
\LEAF.MULTIBIT	45	\OPCODE.SHIFT	45	\SS.OPENING	4
\LEAF.NEVER.OPENED	45	\PASSWORD.ERRORS	44	\SS.TIMEDOUT	4
\LEAF.READBIT	45	\PT.ERROR	45	\SS.UNOPENED	4
\LEAF.WRITEBIT	45	\PT.LEAF	45		
\LEAFMODE.DONTEXTEND	45	\PT.LOOKUPFILE	38		

### VARIABLE INDEX

*UPPER-CASE-FILE-NAMES*	40	LEAFOPCODES	42	SEQUINOPS	3	\LEAF.MAXCACHE	40
DEFAULT.OSTYPE	36	LOOKUPFILECOMS	38	SEQUINSTATES	4	\LEAF.MAXLOOKAHEAD	40
DEFAULTFILETYPE	40	NETWORKOSTYPES	34	SYSTEMRECLST	5, 46	\LEAF.RECOVERY.TIMEOUT	40
IFSERRORS	43	NOFILEPROPEROR	40	UNIXFTPFLG	40	\MAXLEAFTRIES	40
LEAFABORTREGION	40	NONLEAFHOSTS	40	\FTPAVAILABLE	40	\SEQUIN.TIMEOUTMAX	40
LEAFCOMPILETIMECOMS	40	PUPPRINTMACROS	40	\LEAF.CACHETIMEOUT	40	\SOCKET.LEAF	40
LEAFDEBUGFLG	40	SEQUINCOMS	2	\LEAF.IDLETIMEOUT	40		

{MEDLEY}<sources>LEAF.;1

---



---

**RECORD INDEX**

LEAFDATA .....	41	LEAFINFOBLOCK .....	42	LEAFSTREAM .....	42	SEQUIN .....	2
LEAFDEVICE .....	42	LEAFPACKET .....	41	LOOKUPFILEDATA .....	38	SEQUINPACKET .....	2
LEAFERRORDATA .....	41	LEAFPARAMSDATA .....	41	PUPFILESERVER .....	42		

---

**MACRO INDEX**

.NAMEORSTREAM. ....	42	.PAGE.IS.AFTER.EOF. ....	42	INCLEAFSTAT .....	42	SEQUINOP .....	5
---------------------	----	--------------------------	----	-------------------	----	----------------	---

---