```
(RPAQQ FINGERCOMS
        (;; Modified 6-April-87 by smL to interface to TALK; 14-Sep-88 to work in Medley

        (P (IF (BOUNDP 'FINGER.WINDOW)
                THEN
                (END.FINGER)))
        (FNS FINGER REFINGER FINGER.CONTAINS? SEND.FINGER.REQUEST STRING.NOT.NUMERIC)
        (FUNCTIONS NETS.WITHIN)
        [INITVARS (FINGER.TIMEOUT 1000)
                (FINGER.NET.HOPS 2)
                (FINGER.BASE.DATE (IDATE "14-Mar-84 00:00:00"))
                (FINGER.CROWD NIL)
                (FINGER.INFINITY.MINUTES 90)
                (FINGER.CAPABILITIES '(TALK (AND (GETD 'TALK)
                                                  (NOT TALK.GAG)
                                                  (FIND.PROCESS 'COURIER.LISTENER)
                                                  T)
                                        IDLING \IDLING SYSTEM 'Lisp]
        ;; Tablebrowser and window stuff

        (FNS FINGER.SETUP.WINDOW FINGER.MENU.SELECTED FINGER.SETUP.MENU \FINGER.PRINTFN)
        (FILES ICONW TABLEBROWSER)
        (BITMAPS FINGER.ICON.BITMAP FINGER.ICON.MASK FINGER.INFINITY.BITMAP)
        (VARS (FINGER.MENU))
        (INITVARS (FINGER.ICON.POSITION (CREATE POSITION XCOORD _ 900 YCOORD _ 500))
                (FINGER.ICON (ICONW FINGER.ICON.BITMAP FINGER.ICON.MASK FINGER.ICON.POSITION T))
                (FINGER.DISPLAY.WIDTH 290)
                (FINGER.DISPLAY.HEIGHT 140)
                (FINGER.DISPLAY.POSITION (CREATE POSITION XCOORD _ (IDIFFERENCE SCREENWIDTH FINGER.DISPLAY.WIDTH)
                                                  YCOORD _ 0)))
        ;; Responding to finger requests on the net

        (FNS FINGER.SERVER WAIT.FOR.FINGER.PACKET END.FINGER BACKGROUND.FINGER.SERVER)
        ;; Ether info

        (CONSTANTS (FINGER.SERVER.SOCKET# 199)
                (\XIPT.FINGERRESPONSE 20)
                (\XIPT.FINGERREQUEST 21))
        (ALISTS (XIPTYPES \XIPT.FINGERRESPONSE \XIPT.FINGERREQUEST))
        (FNS TRACE.FINGER)
        ;; Start up Finger

        (DECLARE%: DONTEVAL@LOAD DONTEVAL@COMPILE DOCOPY (P (FINGER.SERVER)
                                                          (FINGER)))
        ;; Compiler stuff

        (DECLARE%: EVAL@LOAD DONTCOPY (P (LOADCOMP 'LLNS))
                (FILES TABLEBROWSERDECLS)
                (RECORDS FINGER.HOST)
                (GLOBALVARS FINGER.TIMEOUT FINGER.NET.HOPS FINGER.BASE.DATE FINGER.INFINITY.MINUTES
                        FINGER.ICON.POSITION FINGER.ICON FINGER.ICON.BITMAP FINGER.ICON.MASK
                        FINGER.INFINITY.BITMAP FINGER.DISPLAY.WIDTH FINGER.DISPLAY.HEIGHT FINGER.DISPLAY.POSITION)
                )
        ;; hints to the file manager

        (PROP MAKEFILE-ENVIRONMENT FINGER)))
```

;; Modified 6-April-87 by smL to interface to TALK; 14-Sep-88 to work in Medley

```
(IF (BOUNDP 'FINGER.WINDOW)
    THEN (END.FINGER))
```

```
(DEFINEQ

(FINGER
  [LAMBDA (WHO HOST HOPS ICON?)                                    ; Edited 15-Sep-88 10:43 by smL
    (LET ([USERS (CL:SORT (SEND.FINGER.REQUEST (OR HOPS FINGER.NET.HOPS))
```

```
                          #'(CL:LAMBDA (USER1 USER2)
                                  (CL:STRING< (FETCH (FINGER.HOST USERNAME) OF USER1)
                                              (FETCH (FINGER.HOST USERNAME) OF USER2]
          (WHOM (OR WHO FINGER.CROWD)))
        (if [OR (NOT (BOUNDP 'FINGER.BROWSER))
                (NOT (BOUNDP 'FINGER.WINDOW]
            then (FINGER.SETUP.WINDOW ICON?))
        (if ICON?
            then
```
                 ;; if the icon? is true then just set up the window without opening it or sending a request

```
                (if (NOT (OPENWP FINGER.ICON))
                    then (TOTOPW FINGER.ICON))
          else (WINDOWPROP FINGER.WINDOW 'TITLE (CONCAT "Finger Display at " (SUBSTRING (DATE)
                                                                                        11 15)))
              (TB.REPLACE.ITEMS FINGER.BROWSER)
              [for P in USERS when (COND
                                      ((AND WHOM HOST)
                                       (OR (FINGER.CONTAINS? (CADR P)
                                                        WHOM)
                                           (FINGER.CONTAINS? (CAR P)
                                                        HOST)))
                                      (WHOM (FINGER.CONTAINS? (CADR P)
                                                        WHOM))
                                      (HOST (FINGER.CONTAINS? (CAR P)
                                                        HOST))
                                      (T T))
                   do (TB.INSERT.ITEM FINGER.BROWSER (create TABLEITEM
                                                          TIDATA _ P
                                                          TIUNDELETABLE _ T
                                                          TIUNSELECTABLE _ (NOT (LISTGET (fetch CAPABILITIES
                                                                                            of P)
                                                                                    'TALK]
              (TB.REDISPLAY.ITEMS FINGER.BROWSER])
```

(**REFINGER**
```
  [LAMBDA (W)                                              ; Edited 14-Sep-88 18:03 by smL
```
    ;; dummy fun to call finger w/ no args, so can use as redisplayfn, etc.
```
    (FINGER])
```

(**FINGER.CONTAINS?**
```
  [LAMBDA (ELEMENT L)                                      ; Edited 14-Sep-88 18:03 by smL
```
    ;; returns non-nil if element is list or is contained in list. case-insensitive compare used
```
    (COND
      [(TYPENAMEP L 'LISTP)
       (MEMB (U-CASE ELEMENT)
             (for X in L collect (U-CASE X]
      (T (EQ (U-CASE ELEMENT)
             (U-CASE L])
```

(**SEND.FINGER.REQUEST**
```
  [LAMBDA (NET.HOPS)                                        ; Edited 15-Sep-88 10:44 by smL
   (LET (FINGER.PACKET FINGER.USER.SOCKET NETS RESPONSES UNIQUERESULTS)
        (RESETLST
            [RESETSAVE NIL (LIST 'CLOSENSOCKET (SETQ FINGER.USER.SOCKET (OPENNSOCKET]
```
        ;; Allocate a socket to send on
```
          (SETQ NETS (NETS.WITHIN NET.HOPS))
```
        ;; send this to every one on the net in question TWICE
```
          (for NET in (APPEND NETS NETS) do
```
                                           ;; Get the xip
```
                                              (SETQ FINGER.PACKET (ALLOCATE.XIP))
                                              (\FILLINXIP \XIPT.FINGERREQUEST FINGER.USER.SOCKET
                                                  BROADCASTNSHOSTNUMBER FINGER.SERVER.SOCKET# NET \XIPOVLEN
                                                  FINGER.PACKET)
                                              (SENDXIP FINGER.USER.SOCKET FINGER.PACKET)
                                              (RELEASE.XIP FINGER.PACKET)
                                              (BLOCK))
          (SETQ RESPONSES (for P in (bind PACKET while (SETQ PACKET (GETXIP FINGER.USER.SOCKET FINGER.TIMEOUT)))
                                  collect PACKET)
                          bind PACKET.STREAM DATA
                          collect [SETQ PACKET.STREAM (OPENSTRINGSTREAM (\GETBASESTRING (fetch XIPCONTENTS
                                                                                          of P)
                                                                                  0
                                                                                  (IDIFFERENCE (fetch XIPLENGTH
                                                                                                  of P)
                                                                                          \XIPOVLEN]
                                  [SETQ DATA (CAR (NLSETQ (READ PACKET.STREAM]
                                  (CLOSEF? PACKET.STREAM)
```
                                  ;; don't die on old finger packets when the user was not logged in
```
                                  [if (NUMBERP (CADR DATA))
```

```
                                                then  (SETQ DATA (CONS (CAR DATA)
                                                                  (CONS "[none]" (CDR DATA]
                                DATA))
        ;; The responses are not necessarily eq for the same machine

                    (for DATALIST in RESPONSES when (AND DATALIST (NOT (SASSOC (CAR DATALIST)
                                                                            UNIQUERESULTS)))
                        do (SETQ UNIQUERESULTS (CONS DATALIST UNIQUERESULTS)))
                    UNIQUERESULTS)])
```

(**STRING.NOT.NUMERIC**
```
    [LAMBDA (S)                                              (* edited%: " 3-Aug-84 10:21")
        (AND (STRPOSL (CONSTANT (MAKEBITTABLE (CHCON "0123456789")
                                        T))
                    S)
            S])
```

)

```
(CL:DEFUN NETS.WITHIN (HOPS)
    (DECLARE (GLOBAL \NS.ROUTING.TABLE))
    (LET ((NETS NIL))
        [\MAP.ROUTING.TABLE \NS.ROUTING.TABLE #'(CL:LAMBDA (ENTRY)
                                                    (CL:WHEN (<= (fetch (ROUTING RTHOPCOUNT) of ENTRY)
                                                                HOPS)
                                                        (CL:SETQ NETS (CONS (fetch (ROUTING RTNET#)
                                                                                of ENTRY)
                                                                        NETS)))]
            NETS))
```

(RPAQ? **FINGER.TIMEOUT** 1000)

(RPAQ? **FINGER.NET.HOPS** 2)

(RPAQ? **FINGER.BASE.DATE** (IDATE "14-Mar-84 00:00:00"))

(RPAQ? **FINGER.CROWD** NIL)

(RPAQ? **FINGER.INFINITY.MINUTES** 90)

```
(RPAQ? FINGER.CAPABILITIES '(TALK (AND (GETD 'TALK)
                                        (NOT TALK.GAG)
                                        (FIND.PROCESS 'COURIER.LISTENER)
                                        T)
                                IDLING \IDLING SYSTEM 'Lisp))
```

;; Tablebrowser and window stuff

(DEFINEQ

(**FINGER.SETUP.WINDOW**
```
    [LAMBDA (ICON?)                                                  ; Edited 14-Sep-88 18:04 by smL
        (SETQ FINGER.WINDOW (CREATEW (CREATEREGION (fetch XCOORD of FINGER.DISPLAY.POSITION)
                                                (fetch YCOORD of FINGER.DISPLAY.POSITION)
                                                FINGER.DISPLAY.WIDTH
                                                (HEIGHTIFWINDOW FINGER.DISPLAY.HEIGHT T))
                                    "Finger Display Window" NIL NIL))
        [SETQ FINGER.BROWSER (TB.MAKE.BROWSER NIL FINGER.WINDOW '(PRINTFN \FINGER.PRINTFN]
        (WINDOWPROP FINGER.WINDOW 'ICON FINGER.ICON)
        (WINDOWPROP FINGER.WINDOW 'SCROLLEXTENTUSE (CONS 'LIMIT 'LIMIT))
        (WINDOWADDPROP FINGER.WINDOW 'EXPANDFN 'REFINGER)
        [WINDOWADDPROP FINGER.WINDOW 'CLOSEFN '(LAMBDA NIL
                                                    (SETQ FINGER.WINDOW 'NOBIND]
                                                            ; REFINGER is a dummy fn to call FINGER with no arguments.
        (WINDOWADDPROP FINGER.WINDOW 'RESHAPEFN 'REFINGER)
        (FINGER.SETUP.MENU FINGER.WINDOW FINGER.BROWSER)
        (LET [(FINGER.PROMPT.WINDOW (GETPROMPTWINDOW FINGER.WINDOW 1 (FONTCREATE 'HELVETICA 10]
            [WINDOWPROP FINGER.PROMPT.WINDOW 'MINSIZE (CONS 0 (fetch (REGION HEIGHT) of (WINDOWPROP
                                                                                    FINGER.PROMPT.WINDOW
                                                                                    'REGION]
            [WINDOWPROP FINGER.PROMPT.WINDOW 'MAXSIZE (CONS 64000 (fetch (REGION HEIGHT) of (WINDOWPROP
                                                                                        FINGER.PROMPT.WINDOW
                                                                                        'REGION]
            (LINELENGTH MAX.SMALLP FINGER.PROMPT.WINDOW))
        (if ICON?
            then (SHRINKW FINGER.WINDOW NIL FINGER.ICON.POSITION)
            else                                             ; shouldn't need to open this, but for the moment, one has to
                (OPENW FINGER.ICON)
                (MOVEW FINGER.ICON FINGER.ICON.POSITION)
                (CLOSEW FINGER.ICON])
```

(**FINGER.MENU.SELECTED**
```
    [LAMBDA (ITEM MENU MOUSE)                                        ; Edited 14-Sep-88 18:04 by smL
        (if ITEM
```

```
        then (LET* [(browser (GETMENUPROP MENU 'TB))
                    (promptwindow (GETPROMPTWINDOW (TB.WINDOW browser]
                  (DECLARE%: (SPECVARS promptwindow))
                  (ALLOW.BUTTON.EVENTS)
                  (SHADEITEM ITEM MENU MENUSELECTSHADE)
                  (SELECTQ (CAR ITEM)
                      (Update (FINGER))
                      (Talk (if (GETD 'TALK)
                              then [TB.MAP.SELECTED.ITEMS
                                      browser
                                      [FUNCTION (LAMBDA (browser item)
                                              (DECLARE%: (SPECVARS promptwindow))
                                              (LET ((host (fetch TIDATA of item))
                                                    talkresult)
                                                  (if [NOT (NULL (LISTGET (fetch CAPABILITIES of host)
                                                                      'TALK]
                                                      then (printout promptwindow T "Trying to talk to "
                                                                  (fetch USERNAME of host)
                                                                  "...")
                                                          (SETQ talkresult (TALK (fetch (FINGER.HOST
                                                                                          NSHOSTNUMBER
                                                                                          )
                                                                                  of host)))
                                                          (SELECTQ talkresult
                                                              (T (printout promptwindow T "Talking to "
                                                                      (fetch USERNAME of host)
                                                                      "."))
                                                              (NIL (printout promptwindow T "Talk to "
                                                                      (fetch USERNAME of host)
                                                                      " aborted."))
                                                              (printout promptwindow T talkresult))
                                                      else (printout promptwindow T (fetch USERNAME
                                                                                          of host)
                                                              " on "
                                                              (fetch HOSTNAME of host)
                                                              " is not running TALK."]
                                      (FUNCTION (LAMBDA (browser)
                                              (DECLARE%: (SPECVARS promptwindow))
                                              (printout promptwindow T "No hosts selected to TALK to."]
                              else (printout promptwindow T "Can't -- TALK is not loaded")))
                      NIL)
                  (SHADEITEM ITEM MENU WHITESHADE])
```

(**FINGER.SETUP.MENU**
```
  [LAMBDA (WINDOW TABLE.BROWSER)                                          ; Edited 14-Sep-88 18:04 by smL
    [SETQ FINGER.MENU (create MENU
                          ITEMS _ '((Update NIL "Will update the finger display.")
                                    (Talk NIL "Open a TALK connection to the selected people"))
                          WHENSELECTEDFN _ 'FINGER.MENU.SELECTED
                          CENTERFLG _ T
                          MENUOUTLINESIZE _ (IDIFFERENCE WBorder 3)
                          MENUFONT _ (FONTCREATE '(HELVETICA 8 BOLD]
    (ATTACHMENU FINGER.MENU WINDOW)
    (PUTMENUPROP FINGER.MENU 'TB TABLE.BROWSER])
```

(**FINGER.PRINTFN**
```
  [LAMBDA (browser item window)                                          ; Edited 14-Sep-88 18:04 by smL
    (NLSETQ (LET* ((FINGER.HOST (fetch TIDATA of item))
                   (IDLE (fetch IDLE of FINGER.HOST))
                   (INFINITE.IDLE NIL))                                   ; seconds of idle time
              (if (IGEQ (IQUOTIENT IDLE 60)
                      FINGER.INFINITY.MINUTES)
                  then (SETQ INFINITE.IDLE T)
                  else (SETQ IDLE (SUBSTRING (GDATE (IPLUS FINGER.BASE.DATE IDLE))
                                      11 15)))
              (printout window (if (LISTGET (fetch CAPABILITIES of FINGER.HOST)
                                      'TALK)
                                  then "+"
                                  else " ")
                  (fetch USERNAME of FINGER.HOST)
                  15
                  (L-CASE (fetch HOSTNAME of FINGER.HOST)
                          T)
                  30)
              (if INFINITE.IDLE
                  then (BITBLT FINGER.INFINITY.BITMAP NIL NIL window (DSPXPOSITION NIL window)
                              (DSPYPOSITION NIL window))
                      (DSPXPOSITION (IPLUS (BITMAPWIDTH FINGER.INFINITY.BITMAP)
                                      (DSPXPOSITION NIL window))
                          window)
                  else (PRIN1 IDLE window])
```

```
)
```

```
(FILESLOAD ICONW TABLEBROWSER)
```

(RPAQQ **FINGER.ICON.BITMAP**

)

(RPAQQ **FINGER.ICON.MASK**

)

(RPAQQ **FINGER.INFINITY.BITMAP**)

(RPAQQ **FINGER.MENU** NIL)

(RPAQ? **FINGER.ICON.POSITION** (CREATE POSITION XCOORD _ 900 YCOORD _ 500))

(RPAQ? **FINGER.ICON** (ICONW FINGER.ICON.BITMAP FINGER.ICON.MASK FINGER.ICON.POSITION T))

(RPAQ? **FINGER.DISPLAY.WIDTH** 290)

(RPAQ? **FINGER.DISPLAY.HEIGHT** 140)

(RPAQ? **FINGER.DISPLAY.POSITION** (CREATE POSITION XCOORD _ (IDIFFERENCE SCREENWIDTH FINGER.DISPLAY.WIDTH)
                                                   YCOORD _ 0))

;; Responding to finger requests on the net

(DEFINEQ

(**FINGER.SERVER**
  [LAMBDA NIL   ; Edited 7-Apr-87 15:04 by Briggs

;;; spawn the process which will wait for finger requests, ensuring that there is only one finger server process.

    (DEL.PROCESS (FIND.PROCESS 'Finger% Server))
    (**while** (FIND.PROCESS 'Finger% Server) **do** (BLOCK))   ; wait for the process to really die
    (ADD.PROCESS '(**WAIT.FOR.FINGER.PACKET**)
           'NAME
           'Finger% Server
           'RESTARTABLE
           'HARDRESET)
    NIL])

(**WAIT.FOR.FINGER.PACKET**
  [LAMBDA NIL   (* N.H.Briggs " 7-Apr-87 23:39")

    ;; this function wakes up each time a finger request packet is received, it then sends back etherhostname (or address if no (pup) nameserver) ,the
    ;; username, and the time since the last keyboard or mouse action.

    (LET ((FINGER.SERVER.SOCKET))
         (RESETLST
             (SETQ FINGER.SERVER.SOCKET (OPENNSOCKET FINGER.SERVER.SOCKET# 'ACCEPT))
             (RESETSAVE NIL (LIST 'CLOSENSOCKET FINGER.SERVER.SOCKET T))
             (DISCARDXIPS FINGER.SERVER.SOCKET)
             [**while** T **do** (AWAIT.EVENT (NSOCKETEVENT FINGER.SERVER.SOCKET))
                     (NLSETQ (PROG (RESPONSE.XIP DATA IDLETIME)
                                   (SETQ RESPONSE.XIP (GETXIP FINGER.SERVER.SOCKET))
                                   (**if** (OR (NOT RESPONSE.XIP)
                                          (NEQ (**fetch** XIPTYPE **of** RESPONSE.XIP)
                                               \XIPT.FINGERREQUEST))
                                       then   ; false alarm, go back to sleep
                                            (RETURN))

                          ;; format of response is a string containing a list of the data elements

                                   (**replace** XIPLENGTH **of** RESPONSE.XIP **with** \XIPOVLEN)
                                   (SETQ IDLETIME (IDIFFERENCE (IDATE)
                                                         (ALTO.TO.LISP.DATE \LASTUSERACTION)))
                                   (SETQ DATA (MKSTRING [**create** FINGER.HOST
                                                           HOSTNAME _ (OR (**STRING.NOT.NUMERIC**
                                                                              (ETHERHOSTNAME NIL T))
                                                                          (PORTSTRING (ETHERHOSTNUMBER)))
                                                           USERNAME _
                                                            (**if** (STRING-EQUAL (USERNAME NIL NIL T)
                                                                           "")
                                                                then "[none]"
                                                              else (USERNAME NIL NIL T))
                                                           IDLE _ IDLETIME
                                                           NSHOSTNUMBER _ \MY.NSADDRESS
                                                           CAPABILITIES _
                                                            (**for** CAPABILITY **on** FINGER.CAPABILITIES
                                                               by (CDDR CAPABILITY)

```
                                                              join (LIST (CAR CAPABILITY)
                                                                         (EVAL (CADR CAPABILITY]
                                        T))
                       (XIPAPPEND.STRING RESPONSE.XIP DATA)
                       (SWAPXIPADDRESSES RESPONSE.XIP)
```
                 ;; now have to set the correct source since original dest was nsbroadcastnumber
```
                           (replace XIPTYPE of RESPONSE.XIP with \XIPT.FINGERRESPONSE)
                           (replace XIPSOURCEHOST of RESPONSE.XIP with \MY.NSHOSTNUMBER)
                           (replace XIPSOURCENET of RESPONSE.XIP with 0)
                           (replace XIPSOURCESOCKET of RESPONSE.XIP with FINGER.SERVER.SOCKET#)
                           (SENDXIP FINGER.SERVER.SOCKET# RESPONSE.XIP])])
```

(**END.FINGER**
```
  [LAMBDA NIL                                                  ; Edited  7-Apr-87 14:52 by Briggs
    (DEL.PROCESS (FIND.PROCESS 'Finger% Server))
    (if (BOUNDP 'FINGER.ICON)
        then (if (WINDOWP FINGER.ICON)
                 then (CLOSEW FINGER.ICON)))
    (if (BOUNDP 'FINGER.WINDOW)
        then (if (WINDOWP FINGER.WINDOW)
                 then (CLOSEW FINGER.WINDOW))
             (SETQ FINGER.WINDOW 'NOBIND))
    (if (BOUNDP 'FINGER.BROWSER)
        then (SETQ FINGER.BROWSER 'NOBIND))
    NIL])
```

(**BACKGROUND.FINGER.SERVER**
```
  [LAMBDA NIL                                                  (* N.H.Briggs "15-Apr-87 18:44")
```
    ;; this function wakes up each time a finger request packet is received, it then sends back etherhostname (or address if no (pup) nameserver) ,the
    ;; username, and the time since the last keyboard or mouse action.
```
    (SETQ FINGER.SERVER.SOCKET (OPENNSOCKET FINGER.SERVER.SOCKET# 'ACCEPT))
    (AWAIT.EVENT (NSOCKETEVENT FINGER.SERVER.SOCKET)
           20)
    (NLSETQ (PROG (RESPONSE.XIP DATA IDLETIME)
                (SETQ RESPONSE.XIP (GETXIP FINGER.SERVER.SOCKET))
                (if (OR (NOT RESPONSE.XIP)
                        (NEQ (fetch XIPTYPE of RESPONSE.XIP)
                             \XIPT.FINGERREQUEST))
                    then                                       ; false alarm, go back to sleep
                         (RETURN))
```
             ;; format of response is a string containing a list of the data elements
```
                (replace XIPLENGTH of RESPONSE.XIP with \XIPOVLEN)
                (SETQ IDLETIME (IDIFFERENCE (IDATE)
                                    (ALTO.TO.LISP.DATE \LASTUSERACTION)))
                (SETQ DATA (MKSTRING [create FINGER.HOST
                                         HOSTNAME _ (OR (STRING.NOT.NUMERIC (ETHERHOSTNAME NIL T))
                                                        (PORTSTRING (ETHERHOSTNUMBER)))
                                         USERNAME _ (if (STRING-EQUAL (USERNAME NIL NIL T)
                                                            "")
                                                        then "[none]"
                                                        else (USERNAME NIL NIL T))
                                         IDLE _ IDLETIME
                                         NSHOSTNUMBER _ \MY.NSADDRESS
                                         CAPABILITIES _ (for CAPABILITY on FINGER.CAPABILITIES
                                                             by (CDDR CAPABILITY)
                                                             join (LIST (CAR CAPABILITY)
                                                                        (EVAL (CADR CAPABILITY]
                                        T))
                (XIPAPPEND.STRING RESPONSE.XIP DATA)
                (SWAPXIPADDRESSES RESPONSE.XIP)
```
             ;; now have to set the correct source since original dest was nsbroadcastnumber
```
                (replace XIPTYPE of RESPONSE.XIP with \XIPT.FINGERRESPONSE)
                (replace XIPSOURCEHOST of RESPONSE.XIP with \MY.NSHOSTNUMBER)
                (replace XIPSOURCENET of RESPONSE.XIP with 0)
                (replace XIPSOURCESOCKET of RESPONSE.XIP with FINGER.SERVER.SOCKET#)
                (SENDXIP FINGER.SERVER.SOCKET# RESPONSE.XIP])
)
```

;; Ether info

```
(DECLARE%: EVAL@COMPILE

(RPAQQ FINGER.SERVER.SOCKET# 199)

(RPAQQ \XIPT.FINGERRESPONSE 20)

(RPAQQ \XIPT.FINGERREQUEST 21)

(CONSTANTS (FINGER.SERVER.SOCKET# 199)
       (\XIPT.FINGERRESPONSE 20)
```

```
        (\XIPT.FINGERREQUEST 21))
)

(ADDTOVAR XIPTYPES (\XIPT.FINGERRESPONSE 20)
                   (\XIPT.FINGERREQUEST 21))

(DEFINEQ
```

(TRACE.FINGER
```
  [LAMBDA NIL                                                    (* smL " 6-Apr-87 17:17")
    (SETQ XIPONLYTYPES (LIST \XIPT.FINGERREQUEST \XIPT.FINGERRESPONSE))
    (XIPTRACE T])
)
```

;; Start up Finger

```
(DECLARE%: DONTEVAL@LOAD DONTEVAL@COMPILE DOCOPY
```

(**FINGER.SERVER**)

(**FINGER**)
```
)
```

;; Compiler stuff

```
(DECLARE%: EVAL@LOAD DONTCOPY

(LOADCOMP 'LLNS)

(FILESLOAD TABLEBROWSERDECLS)

(DECLARE%: EVAL@COMPILE

(RECORD FINGER.HOST (HOSTNAME USERNAME IDLE NSHOSTNUMBER CAPABILITIES))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS FINGER.TIMEOUT FINGER.NET.HOPS FINGER.BASE.DATE FINGER.INFINITY.MINUTES FINGER.ICON.POSITION
        FINGER.ICON FINGER.ICON.BITMAP FINGER.ICON.MASK FINGER.INFINITY.BITMAP FINGER.DISPLAY.WIDTH
        FINGER.DISPLAY.HEIGHT FINGER.DISPLAY.POSITION)
)
)
```

;; hints to the file manager

```
(PUTPROPS FINGER MAKEFILE-ENVIRONMENT (:READTABLE "INTERLISP" :PACKAGE "INTERLISP"))
```

```
(PUTPROPS FINGER COPYRIGHT ("Xerox Corporation" 1985 1987 1988))
```

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

PROPERTY INDEX

RECORD INDEX