

File created: 14-Oct-2025 10:02:04 {WMEDLEY}<lispusers>talk>TALK-IP.;4

edit by: rmk

previous date: 14-Oct-2025 10:01:37 {WMEDLEY}<lispusers>talk>TALK-IP.;3

Read Table: INTERLISP

Package: INTERLISP

Format: MCCS

```
(RPAQQ TALK-IPCOMS
  (* TALK (Interim)
    IP Interface)
  (LOCALVARS . T)
  (FNS TALK.IP.SERVER)
  (FNS TALK.IP.USERNAME TALK.IP.CONNECT TALK.IP.EVENT TALK.START.IP.SERVER)
  (INITVARS (TALK.UDP.PORT 517))
  (GLOBALVARS TALK.UDP.PORT TALK.IP.CONSTANTS)
  (DECLARE%: DONTCOPY (RECORDS TALK.IP.PACKET)
    (CONSTANTS * TALK.IP.CONSTANTS))
  (* etc)
  (FILES TALK TCP TCPUdp)
  (APPENDVARS (TALK.PROTOCOLTYPES (IP DODIP.HOSTP TALK.IP.USERNAME TALK.IP.CONNECT TALK.IP.EVENT
    TALK.START.IP.SERVER)))
  (DECLARE%: DONTCOPY DONTVAL@LOAD EVAL@COMPILE (FILES ETHERRECORDS TCPEXPORTS))
  (P (TALK.START.IP.SERVER))))

(* * TALK (Interim) IP Interface)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DEFINEQ
TALK.IP.SERVER
[LAMBDA NIL
  (DECLARE (GLOBALVARS \IP.READY))
  (LET (SOCKET)
    (DECLARE (SPECVARS SOCKET))
    (RESETLST
      [RESETSAVE NIL '(UDP.CLOSE.SOCKET ,(SETQ SOCKET (UDP.OPEN.SOCKET TALK.UDP.PORT]
      [bind PACKET RESPONSE SERVICE GAP.SERVICETYPE TALK.SERVICETYPE INPUTSTREAM
      OUTPUTSTREAM PORT USER while \IP.READY do (SETQ PACKET (UDP.GET SOCKET T))
      (UDP.SETUP (SETQ RESPONSE (
        ALLOCATE.ETHERPACKET
        ))
        (with IP PACKET IPSOURCEADDRESS)
        (with UDP PACKET UDPSOURCEPORT)
        0 SOCKET 'FREE)
      (UDP.APPEND.BYTE RESPONSE (with
        TALK.IP.PACKET
        PACKET
        TALK.SERVICE.BYTE
        ))
      (if [OR [NULL (if (SETQ GAP.SERVICETYPE
        (ASSOC (with TALK.IP.PACKET
        PACKET
        TALK.SERVICE.BYTE
        )
        GAP.SERVICETYPES))
        then (SETQ SERVICE
        (with GAP.SERVICETYPE
        GAP.SERVICETYPE
        GAP.SERVICENAME
        )
        (NULL (SETQ TALK.SERVICETYPE
        (ASSOC SERVICE
        TALK.SERVICETYPES]
        then (UDP.APPEND.BYTE RESPONSE
        \IPTALK.NOSERVICE)
        (UDP.SEND SOCKET RESPONSE)
```

```

elseif [OR TALK.GAG (NOT (TALK.ANSWER
    (SETQ USER
      (with
        TALK.IP.PACKET
        PACKET
        TALK.IP.USERNAME
      ))
      SERVICE
      'IP
      (with IP PACKET
        IPSOURCEADDRESS
      )
    then (UDP.APPEND.BYTE RESPONSE
      \IPTALK.NOANSWER)
      (UDP.SEND SOCKET RESPONSE)
    else (UDP.APPEND.BYTE RESPONSE
      \IPTALK.SUCCESS)
      (UDP.APPEND.WORD RESPONSE
        (SETQ PORT (\TCP.SELECT.PORT
          )))
      (UDP.SEND SOCKET RESPONSE)
      (if (SETQ INPUTSTREAM (TCP.OPEN
        (with IP PACKET
          IPSOURCEADDRESS
        )
        NIL PORT
        'PASSIVE
        'INPUT))
        then (TALK.PROCESS
          INPUTSTREAM (
            TCP.OTHER.STREAM
            INPUTSTREAM
          )
          TALK.SERVICETYPE
          'IP
          'SERVER USER T)]])
)

```

(DEFINEQ

(TALK.IP.USERNAME

[LAMBDA (INPUTSTREAM OUTPUTSTREAM SERVICETYPE MODE USER)

; Edited 8-Jun-88 15:45 by cdl

(SELECTQ (with TALK.SERVICETYPE SERVICETYPE TALK.SERVICENAME)

((TTY Sketch)

(* For (backward) compatibility)

USER)

(LET ((NAME (USERNAME)))

(PRINTOUT OUTPUTSTREAM (if (NOT (STREQUAL NAME (CONSTANT null)))
then NAME)

T)

(FORCEOUTPUT OUTPUTSTREAM)

(SETQ NAME (RATOM INPUTSTREAM TALK.READTABLE))

(* Eat EOL)

(BIN INPUTSTREAM)

(OR NAME USER))

(TALK.IP.CONNECT

[LAMBDA (HOST SERVICETYPES)

; Edited 13-Jun-88 17:54 by cdl

(DECLARE (SPECVARS HOST SERVICETYPES))

(LET (SOCKET)

(DECLARE (SPECVARS SOCKET))

(RESETLST

[RESETSAVE NIL '(UDP.CLOSE.SOCKET ,(SETQ SOCKET (UDP.OPEN.SOCKET)

[PROG (NAME REQUEST RESPONSE INPUTSTREAM RESULT)

(while (STREQUAL (SETQ NAME (USERNAME))

(CONSTANT null))

```

do (LOGIN))
(if [LITATOM (SETQ RESULT (for SERVICETYPE in SERVICETYPES
                           thereis (PROGN (UDP.SETUP (SETQ REQUEST (
                                                                \ALLOCATE.ETHERPACKET
                                                                ))
                                                                HOST TALK.UDP.PORT 0 SOCKET 'FREE)
                                                                (UDP.APPEND.BYTE REQUEST
                                                                (with GAP.SERVICETYPE
                                                                [for GAP.SERVICETYPE in
                                                                GAP.SERVICETYPES
                                                                thereis (with GAP.SERVICETYPE
                                                                GAP.SERVICETYPE
                                                                (with TALK.SERVICETYPE
                                                                SERVICETYPE
                                                                (EQ
                                                                GAP.SERVICENAME
                                                                TALK.SERVICENAME
                                                                ]
                                                                GAP.UNSPECIFIED))
                                                                (UDP.APPEND.BYTE REQUEST 0)
                                                                (UDP.APPEND.WORD REQUEST 0)
                                                                (UDP.APPEND.WORD REQUEST (NCHARS NAME))
                                                                (UDP.APPEND.STRING REQUEST NAME)
                                                                (if [SETQ RESPONSE (UDP.EXCHANGE
                                                                SOCKET REQUEST
                                                                (TIMES TALK.ANSWER.WAIT
                                                                (CONSTANT (PROGN
                                                                (* Convert to milliseconds and double in case they are idle)
                                                                (TIMES 2 1000
                                                                ]
                                                                then (SELECT (with TALK-IP-PACKET RESPONSE
                                                                TALK-STATUS)
                                                                (\IPTALK-SUCCESS T)
                                                                (\IPTALK-NOSERVICE NIL)
                                                                (\IPTALK-NOANSWER (RETURN
                                                                'ANSWER))
                                                                (RETURN 'CONNECT))
                                                                else
                                                                (* Can't connect)
                                                                (RETURN 'CONNECT]
                                                                then (RETURN RESULT)
                                                                else (if (STREAMP (SETQ INPUTSTREAM (TCP.OPEN HOST (with TALK-IP-PACKET RESPONSE
                                                                TALK-TEDIT-PORT)
                                                                NIL
                                                                'ACTIVE
                                                                'INPUT T)))
                                                                then [RETURN (CONS RESULT (CONS INPUTSTREAM (TCP.OTHER-STREAM
                                                                INPUTSTREAM]
                                                                else (RETURN 'CONNECT]))))

```

```

(TALK-IP.EVENT
[LAMBDA (INPUTSTREAM OUTPUTSTREAM)
  (while (AND (OPENP INPUTSTREAM)
              (OPENP OUTPUTSTREAM)
              (NOT (READP INPUTSTREAM))))
  do (if (EOFP INPUTSTREAM)
        then (CLOSEF? INPUTSTREAM))
      (BLOCK]))

```

(* cdl "18-May-87 16:29")

(TALK-START-IP-SERVER

[LAMBDA (RESTART)

; Edited 17-Jun-88 12:20 by cdl

[LET [(DEVICE (\GETDEVICEFROMNAME 'TCP 'NOERROR 'DONTCREATE]

(if DEVICE

then

(* (Temporary) patch to make TCP streams handle NS character codes)

(with FDEV DEVICE (if (NULL READCHARCODE)

then (SETQ READCHARCODE (FUNCTION \GENERIC.READCCODE])

```
(bind PROCESS while (AND (SETQ PROCESS (FIND.PROCESS 'TALK.IP.SERVER))
                          RESTART)
  do (DEL.PROCESS PROCESS)
    (BLOCK)
    yield
    (if PROCESS
      then PROCESS
      elseif \IP.READY
      then (ADD.PROCESS '(TALK.IP.SERVER)
                        'RESTARTABLE
                        'SYSTEM))

)

(RPAQ? TALK.UDP.PORT 517)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TALK.UDP.PORT TALK.IP.CONSTANTS)
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

[ACCESSFNS TALK.IP.PACKET [(TALK.PACKET.BASE (with UDP DATUM UDPCONTENTS))
                          (TALK.IP.USERNAME (\GETBASESTRING (with UDP DATUM UDPCONTENTS)
                                                             6
                                                             (with TALK.IP.PACKET DATUM TALK.USERNAME.LENGTH])
                          (BLOCKRECORD TALK.PACKET.BASE ((TALK.SERVICE.BYTE BYTE)
                                                          (TALK.STATUS BYTE)
                                                          (TALK.TEDIT.PORT WORD)
                                                          (TALK.USERNAME.LENGTH WORD)
                          )

(RPAQQ TALK.IP.CONSTANTS ((\IPTALK.SUCCESS 0)
                          (\IPTALK.NOSERVICE 1)
                          (\IPTALK.NOANSWER 2)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \IPTALK.SUCCESS 0)

(RPAQQ \IPTALK.NOSERVICE 1)

(RPAQQ \IPTALK.NOANSWER 2)

(CONSTANTS (\IPTALK.SUCCESS 0)
            (\IPTALK.NOSERVICE 1)
            (\IPTALK.NOANSWER 2))
)

)

(* * etc)

(FILESLoad TALK TCP TCPUDP)

(APPENDTOVAR TALK.PROTOCOLTYPES (IP DODIP.HOSTP TALK.IP.USERNAME TALK.IP.CONNECT TALK.IP.EVENT TALK.START.IP.SERVER
))

(DECLARE%: DONTCOPY DONTVAL@LOAD EVAL@COMPILE

(FILESLoad ETHERRECORDS TCPEXPORTS)
)

(TALK.START.IP.SERVER)
```

FUNCTION INDEX

TALK.IP.CONNECT	2	TALK.IP.SERVER	1	TALK.START.IP.SERVER	3
TALK.IP.EVENT	3	TALK.IP.USERNAME	2		

CONSTANT INDEX

\IPTALK.NOANSWER	4	\IPTALK.NOSERVICE	4	\IPTALK.SUCCESS	4
------------------------	---	-------------------------	---	-----------------------	---

VARIABLE INDEX

TALK.IP.CONSTANTS	4	TALK.UDP.PORT	4
-------------------------	---	---------------------	---

RECORD INDEX

TALK.IP.PACKET	4
----------------------	---
