```
(RPAQQ TCPCHATCOMS
       [(FNS TCPCHAT.BIN TCPCHAT.HOST.FILTER TCPCHAT.NEGOTIATE TCPCHAT.OPEN TCPCHAT.OPTION.COMMAND
             TCPCHAT.OPTION.INPUT TCPCHAT.OPTION.OUTPUT TCPCHAT.OPTION.TRACE TCPCHAT.TERMINAL.TYPE)
        (VARS TCPCHAT.TELNET.TTY.TYPES TELNET.OPTIONS)
        (INITVARS (TCPCHAT.TRACEFLG)
              (TCPCHAT.TRACEFILE))
        (FILES (SYSLOAD)
              TCP CHAT)
        (DECLARE%: EVAL@COMPILE DONTCOPY (FILES CHATDECLS)
              (COMS (CONSTANTS * TELNET.COMMANDS)
                    (CONSTANTS * TELNET.MARKS))
              (RECORDS TELNET.OPTION TELNET.OPTIONSTATE)
              (GLOBALVARS TCPCHAT.TELNET.TTY.TYPES TELNET.OPTIONS TELNET.MARKS))
        (DECLARE%: DONTEVAL@LOAD DOCOPY

              ;; Tell Chat we exist

              (ADDVARS (CHAT.PROTOCOLTYPES (TCP . TCPCHAT.HOST.FILTER))
                    (CHAT.PROTOCOL.ABBREVS (N . TCP)
                          (T . TCP])

(DEFINEQ
```

(**TCPCHAT.BIN**
```
  [LAMBDA (STREAM)                                          ; Edited  7-Jul-88 18:03 by atm
    (bind CHAR while (AND (EQ (SETQ CHAR (\BUFFERED.BIN STREAM))
                             TELNET.IAC)
                         (NEQ (SETQ CHAR (\BUFFERED.BIN STREAM))
                             TELNET.IAC))
       do (TCPCHAT.NEGOTIATE CHAR STREAM) finally (RETURN CHAR])
```

(**TCPCHAT.HOST.FILTER**
```
  [LAMBDA (HOST)                                            ; Edited 12-Apr-88 17:14 by bvm
    (COND
      ((AND \IPFLG (DODIP.HOSTP HOST))
       (LIST HOST (FUNCTION TCPCHAT.OPEN])
```

(**TCPCHAT.NEGOTIATE**
```
  [LAMBDA (COMMAND STREAM)                                  ; Edited  7-Jul-88 18:03 by atm
    (TCPCHAT.OPTION.INPUT (TCP.OTHER.STREAM STREAM)
          COMMAND
          (\BUFFERED.BIN STREAM])
```

(**TCPCHAT.OPEN**
```
  [LAMBDA (HOST)                                            ; Edited 17-Apr-87 10:06 by jrb:
    (PROG ((STREAM (TCP.OPEN (DODIP.HOSTP HOST)
                         \TCP.TELNET.PORT NIL 'ACTIVE 'INPUT))
           [OSTYPE (OR (AND (GETHASH (U-CASE HOST)
                                  \IP.HOSTNAMES)
                            (fetch (HOSTS.TXT.ENTRY HTE.OS.TYPE) of (GETHASH (U-CASE HOST)
                                                                             \IP.HOSTNAMES)))
                       (GETHOSTINFO HOST 'OSTYPE]
           OUTPUTSTREAM)
          (COND
            (STREAM (replace (STREAM BINABLE) of STREAM with NIL) ; Can't run microcoded
                    (replace (STREAM STRMBINFN) of STREAM with (FUNCTION TCPCHAT.BIN))
                    (STREAMPROP STREAM 'SETDISPLAYTYPE (FUNCTION NILL))
                    [COND
                      ((EQ OSTYPE 'INTERLISP)
                       (RETURN (CONS STREAM (TCP.OTHER.STREAM STREAM]
                                                                   ; (STREAMPROP STREAM (QUOTE SETDISPLAYTYPE)
                                                                   ; (FUNCTION TCPCHAT.TERMINAL.TYPE))
                    (SETQ OUTPUTSTREAM (TCP.OTHER.STREAM STREAM))
                    [STREAMPROP OUTPUTSTREAM 'OPTIONSTATES (for OPTION in TELNET.OPTIONS
                                                             collect (create TELNET.OPTIONSTATE
                                                                        OPTION _ (fetch (TELNET.OPTION OPTION)
                                                                                     of OPTION]
                    (TCPCHAT.OPTION.COMMAND OUTPUTSTREAM TELNET.DO TELNET.ECHO)
```

```
                       (TCPCHAT.OPTION.COMMAND OUTPUTSTREAM TELNET.DO TELNET.SUPPRESS.GOAHEAD)
                       (TCPCHAT.OPTION.COMMAND OUTPUTSTREAM TELNET.WILL TELNET.SUPPRESS.GOAHEAD)
                       (COND
                          ((NEQ OSTYPE 'UNIX)
                           (TCPCHAT.OPTION.COMMAND OUTPUTSTREAM TELNET.DO TELNET.BINARY)
                           (TCPCHAT.OPTION.COMMAND OUTPUTSTREAM TELNET.WILL TELNET.BINARY)))
                       (RETURN (CONS STREAM OUTPUTSTREAM])
```

(**TCPCHAT.OPTION.COMMAND**
```
  [LAMBDA (OUTPUTSTREAM COMMAND OPTION TRACECAPTION)                       ; Edited 24-Aug-87 16:58 by scp
    (LET ([OPTIONSTATE (FASSOC OPTION (STREAMPROP OUTPUTSTREAM 'OPTIONSTATES]
          GO.AHEAD.WITH.COMMAND)
         (COND
            ((NULL OPTIONSTATE)
             (SETQ GO.AHEAD.WITH.COMMAND T))
            (T (SELECTC COMMAND
                   (TELNET.DO [COND
                                 ((NEQ (fetch (TELNET.OPTIONSTATE DOING) of OPTIONSTATE)
                                       'YES)
                                  (SETQ GO.AHEAD.WITH.COMMAND T)
                                  (replace (TELNET.OPTIONSTATE DOING) of OPTIONSTATE with 'YES])
                   (TELNET.WILL [COND
                                 ((NEQ (fetch (TELNET.OPTIONSTATE WILLING) of OPTIONSTATE)
                                       'YES)
                                  (SETQ GO.AHEAD.WITH.COMMAND T)
                                  (replace (TELNET.OPTIONSTATE WILLING) of OPTIONSTATE with 'YES])
                   (TELNET.DONT [COND
                                 ((NEQ (fetch (TELNET.OPTIONSTATE DOING) of OPTIONSTATE)
                                       'NO)
                                  (SETQ GO.AHEAD.WITH.COMMAND T)
                                  (replace (TELNET.OPTIONSTATE DOING) of OPTIONSTATE with 'NO])
                   (TELNET.WONT [COND
                                 ((NEQ (fetch (TELNET.OPTIONSTATE WILLING) of OPTIONSTATE)
                                       'NO)
                                  (SETQ GO.AHEAD.WITH.COMMAND T)
                                  (replace (TELNET.OPTIONSTATE WILLING) of OPTIONSTATE with 'NO])
                   NIL)))
         (COND
            (GO.AHEAD.WITH.COMMAND (BOUT OUTPUTSTREAM TELNET.IAC)
                   (BOUT OUTPUTSTREAM COMMAND)
                   (BOUT OUTPUTSTREAM OPTION)
                   (FORCEOUTPUT OUTPUTSTREAM)
                   (TCPCHAT.OPTION.TRACE COMMAND OPTION (OR TRACECAPTION 'SEND])
```

(**TCPCHAT.OPTION.INPUT**
```
  [LAMBDA (OUTPUTSTREAM COMMAND OPTION)                               ; Edited 16-Apr-87 13:30 by jrb:
    (LET ((OPTIONRECORD (FASSOC OPTION TELNET.OPTIONS)))
         (COND
            (OPTIONRECORD (SELECTC COMMAND
                            (TELNET.DO (TCPCHAT.OPTION.TRACE 'DO OPTION 'RECV)
                                   (TCPCHAT.OPTION.OUTPUT OUTPUTSTREAM (fetch (TELNET.OPTION ON.DO)
                                                                            of OPTIONRECORD)
                                         OPTION))
                            (TELNET.DONT (TCPCHAT.OPTION.TRACE 'DONT OPTION 'RECV)
                                   (TCPCHAT.OPTION.OUTPUT OUTPUTSTREAM (fetch (TELNET.OPTION ON.DONT)
                                                                            of OPTIONRECORD)
                                         OPTION))
                            (TELNET.WILL (TCPCHAT.OPTION.TRACE 'WILL OPTION 'RECV)
                                   (TCPCHAT.OPTION.OUTPUT OUTPUTSTREAM (fetch (TELNET.OPTION ON.WILL)
                                                                            of OPTIONRECORD)
                                         OPTION))
                            (TELNET.WONT (TCPCHAT.OPTION.TRACE 'WONT OPTION 'RECV)
                                   (TCPCHAT.OPTION.OUTPUT OUTPUTSTREAM (fetch (TELNET.OPTION ON.WONT)
                                                                            of OPTIONRECORD)
                                         OPTION))
                            (TELNET.SB (TCPCHAT.OPTION.TRACE 'SB OPTION 'RECV)
                                   (TCPCHAT.OPTION.OUTPUT OUTPUTSTREAM (fetch (TELNET.OPTION ON.SB)
                                                                            of OPTIONRECORD)
                                         OPTION))
                            COMMAND))
            (T (TCPCHAT.OPTION.TRACE COMMAND OPTION)
               (TCPCHAT.OPTION.OUTPUT OUTPUTSTREAM 'WONT OPTION])
```

(**TCPCHAT.OPTION.OUTPUT**
```
  [LAMBDA (OUTPUTSTREAM COMMAND OPTION)                               ; Edited 17-Apr-87 16:34 by jrb:
    (LET (CMDNUM)
         (COND
            ((NULL COMMAND))
            ([SETQ CMDNUM (CDR (FASSOC COMMAND '((WILL \, TELNET.WILL)
                                                (WONT \, TELNET.WONT)
                                                (DO \, TELNET.DO)
                                                (DONT \, TELNET.DONT]
             (TCPCHAT.OPTION.COMMAND OUTPUTSTREAM CMDNUM OPTION 'SENDBACK))
            (T (APPLY* COMMAND (TCP.OTHER.STREAM OUTPUTSTREAM])
```

(TCPCHAT.OPTION.TRACE
```
  [LAMBDA (COMMAND OPTION PREFIX)                                        (* ejs%: "22-Apr-85 16:41")
    (DECLARE (GLOBALVARS TCPCHAT.TRACEFLG TCPCHAT.TRACEFILE))
    (COND
       (TCPCHAT.TRACEFLG [COND
                             ((SMALLP COMMAND)
                              (SETQ COMMAND (SELECTC COMMAND
                                                    (TELNET.DO 'DO)
                                                    (TELNET.DONT 'DONT)
                                                    (TELNET.WILL 'WILL)
                                                    (TELNET.WONT 'WONT)
                                                    COMMAND]
                        (printout TCPCHAT.TRACEFILE PREFIX ": " COMMAND " ")
                        (PRINTCONSTANT OPTION TELNET.MARKS TCPCHAT.TRACEFILE)
                        (TERPRI TCPCHAT.TRACEFILE])
```

(TCPCHAT.TERMINAL.TYPE
```
  [LAMBDA (INPUTSTREAM)                                                  ; Edited 20-Apr-87 13:42 by jrb:
    (LET ((COMMAND))
        (SELECTC (\BUFFERED.BIN INPUTSTREAM)
              (TELNET.SEND                                               ; OK, should be followed by IAC SE
                        (IF TCPCHAT.TRACEFLG
                            THEN (printout TCPCHAT.TRACEFILE "REQUEST IS SEND"))
                        (IF (EQ (SETQ COMMAND (\BUFFERED.BIN INPUTSTREAM))
                                TELNET.IAC)
                            THEN (IF TCPCHAT.TRACEFLG
                                     THEN (printout TCPCHAT.TRACEFILE " IAC"))
                          ELSE (IF TCPCHAT.TRACEFLG
                                   THEN (printout " EXPECTED IAC, GOT " COMMAND)))
                        (IF (EQ (SETQ COMMAND (\BUFFERED.BIN INPUTSTREAM))
                                TELNET.SE)
                            THEN (IF TCPCHAT.TRACEFLG
                                     THEN (printout TCPCHAT.TRACEFILE " SE"))
                          ELSE (IF TCPCHAT.TRACEFLG
                                   THEN (printout " EXPECTED SE, GOT " COMMAND)))
                        (IF TCPCHAT.TRACEFLG
                            THEN (TERPRI TCPCHAT.TRACEFILE))
                        [LET* [(OUTPUTSTREAM (TCP.OTHER.STREAM INPUTSTREAM))
                               (DISPLAYTYPE (OR (CDR (FASSOC (fetch (CHATDISPLAYTYPE DPYNAME)
                                                                of (STREAMPROP INPUTSTREAM 'DISPLAYTYPE))
                                                             TCPCHAT.TELNET.TTY.TYPES))
                                                (CDR (FASSOC (fetch (CHATDISPLAYTYPE DPYNAME)
                                                                of (STREAMPROP OUTPUTSTREAM 'DISPLAYTYPE))
                                                             TCPCHAT.TELNET.TTY.TYPES]
                             (BOUT OUTPUTSTREAM TELNET.IAC)
                             (BOUT OUTPUTSTREAM TELNET.SB)
                             (BOUT OUTPUTSTREAM TELNET.TERMINAL.TYPE)
                             (BOUT OUTPUTSTREAM TELNET.IS)
                             (PRIN1 DISPLAYTYPE OUTPUTSTREAM)
                             (BOUT OUTPUTSTREAM TELNET.IAC)
                             (BOUT OUTPUTSTREAM TELNET.SE)
                             (FORCEOUTPUT OUTPUTSTREAM)
                             (COND
                                (TCPCHAT.TRACEFLG (printout TCPCHAT.TRACEFILE "SEND(BACK) IAC SB
                                                            TERMINAL-TYPE IS " DISPLAYTYPE " IAC SE" T])
              (TELNET.IS                                                 ; We told them we couldn't handle this - or would have had they
                                                                         ; asked...
                        (IF TCPCHAT.TRACEFLG
                            THEN (printout TCPCHAT.TRACEFILE "REQUEST IS IS, which is an error: rest of command
                                           is:" T))
                        (WHILE (NEQ (SETQ COMMAND (\BUFFERED.BIN INPUTSTREAM))
                                    TELNET.SE)
                           DO (IF TCPCHAT.TRACEFLG
                                  THEN (PRIN1 (CHARACTER COMMAND)
                                              TCPCHAT.TRACEFILE)))
                        (IF TCPCHAT.TRACEFLG
                            THEN (printout TCPCHAT.TRACEFILE " SE" T)))
              (IF TCPCHAT.TRACEFLG
                  THEN (printout TCPCHAT.TRACEFILE "REQUEST IS " COMMAND ", which is an error" T])
)
```

```
(RPAQQ TCPCHAT.TELNET.TTY.TYPES ((DM2500 . DATAMEDIA-2500)
                                 (VT100 . DEC-VT100)))
```

```
(RPAQQ TELNET.OPTIONS
       ((94 WONT WONT DONT DONT)
        (0 WILL WONT NIL DONT)
        (1 WONT WONT DO DO)
        (3 WILL WILL NIL NIL)
        (5 WONT WONT DONT DONT)
        (6 WILL NIL NIL NIL)
        (24 WILL NIL DONT NIL TCPCHAT.TERMINAL.TYPE)))
```

```
(RPAQ? TCPCHAT.TRACEFLG )

(RPAQ? TCPCHAT.TRACEFILE )

(FILESLOAD (SYSLOAD)
        TCP CHAT)

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD CHATDECLS)

(RPAQQ TELNET.COMMANDS ((TELNET.SE 240)
                        (TELNET.SB 250)
                        (TELNET.WILL 251)
                        (TELNET.WONT 252)
                        (TELNET.DO 253)
                        (TELNET.DONT 254)
                        (TELNET.IAC 255)
                        (TELNET.SEND 1)
                        (TELNET.IS 0)))

(DECLARE%: EVAL@COMPILE

(RPAQQ TELNET.SE 240)

(RPAQQ TELNET.SB 250)

(RPAQQ TELNET.WILL 251)

(RPAQQ TELNET.WONT 252)

(RPAQQ TELNET.DO 253)

(RPAQQ TELNET.DONT 254)

(RPAQQ TELNET.IAC 255)

(RPAQQ TELNET.SEND 1)

(RPAQQ TELNET.IS 0)

(CONSTANTS (TELNET.SE 240)
        (TELNET.SB 250)
        (TELNET.WILL 251)
        (TELNET.WONT 252)
        (TELNET.DO 253)
        (TELNET.DONT 254)
        (TELNET.IAC 255)
        (TELNET.SEND 1)
        (TELNET.IS 0))
)

(RPAQQ TELNET.MARKS ((TELNET.BINARY 0)
                     (TELNET.ECHO 1)
                     (TELNET.SUPPRESS.GOAHEAD 3)
                     (TELNET.STATUS 5)
                     (TELNET.TIMING.MARK 6)
                     (TELNET.TERMINAL.TYPE 24)))

(DECLARE%: EVAL@COMPILE

(RPAQQ TELNET.BINARY 0)

(RPAQQ TELNET.ECHO 1)

(RPAQQ TELNET.SUPPRESS.GOAHEAD 3)

(RPAQQ TELNET.STATUS 5)

(RPAQQ TELNET.TIMING.MARK 6)

(RPAQQ TELNET.TERMINAL.TYPE 24)

(CONSTANTS (TELNET.BINARY 0)
        (TELNET.ECHO 1)
        (TELNET.SUPPRESS.GOAHEAD 3)
        (TELNET.STATUS 5)
        (TELNET.TIMING.MARK 6)
        (TELNET.TERMINAL.TYPE 24))
)

(DECLARE%: EVAL@COMPILE

(RECORD TELNET.OPTION (OPTION ON.DO ON.DONT ON.WILL ON.WONT ON.SB))

(RECORD TELNET.OPTIONSTATE (OPTION WILLING DOING))
)
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TCPCHAT.TELNET.TTY.TYPES TELNET.OPTIONS TELNET.MARKS)
)
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(ADDTOVAR CHAT.PROTOCOLTYPES (TCP . TCPCHAT.HOST.FILTER))

(ADDTOVAR CHAT.PROTOCOL.ABBREVS (N . TCP)
                                (T . TCP))
)

(PUTPROPS TCPCHAT COPYRIGHT ("Venue & Xerox Corporation" 1985 1986 1987 1988 1990))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX