changes to: (FNS \AR.DAEMON \AR.ENTER.RESOLUTION \AR.NOTE.RESOLUTION \AR.UPDATE.RESOLUTION \PRINTAR SPUTASSOC
          \AR.TRANSLATE.TO.10MB \AR.REQUEST.IP.TO.10MB \AR.REQUEST.IP.TO.3MB \AR.RESOLVE
          \AR.TRANSLATE.TO.3MB \HANDLE.RAW.AR)

previous date: 6-Jan-89 15:18:06   {DSK}<LISPFILES>ETHERNET>TCP>NEW>TCPLLAR.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1985, 1986, 1987, 1988, 1989, 1990 by Xerox Corporation.  All rights reserved.


(RPAQQ TCPLLARCOMS
       [ (COMS

;;;; IP Ethernet address translation module

                [DECLARE%: DONTCOPY (EXPORT (RECORDS AR ARETHER AREXPETHER ARENTRY)
                                        (CONSTANTS (\AR.HARDWARE.SPACE.ETHERNET 1)
                                                (\AR.ETHERNET.ADDRESS.LENGTH 6)
                                                (\AR.IP.ADDRESS.LENGTH 4)
                                                (\AR.REQUEST 1)
                                                (\AR.RESPONSE 2)
                                                (\AR.ETHER.PACKET.LENGTH 28]
                (INITRECORDS ARENTRY)
                (INITVARS (\AR.IP.TO.10MB.ALIST (CONS))
                        (\AR.SEARCH.TIMEOUT.INTERVAL 300000)
                        (\AR.VALID.TIMEOUT.INTERVAL 600000))
                (GLOBALVARS \AR.IP.TO.10MB.ALIST \AR.SEARCH.TIMEOUT.INTERVAL \AR.VALID.TIMEOUT.INTERVAL)
                (FNS \AR.DAEMON \AR.ENTER.RESOLUTION \AR.NOTE.RESOLUTION \AR.UPDATE.RESOLUTION \PRINTAR SPUTASSOC
                        \AR.TRANSLATE.TO.10MB \AR.REQUEST.IP.TO.10MB \AR.REQUEST.IP.TO.3MB \AR.RESOLVE
                        \AR.TRANSLATE.TO.3MB \HANDLE.RAW.AR)
                (ADDVARS (\PACKET.PRINTERS (2054 . \PRINTAR])


;;;; IP Ethernet address translation module

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

[ACCESSFNS AR [(ARBASE (LOCF (fetch (ETHERPACKET EPBODY) of DATUM]
        (BLOCKRECORD ARBASE ((ARHARDWARESPACE WORD)
                                (ARPROTOCOLSPACE WORD)
                                (ARHARDWARELEN BYTE)
                                (ARPROTOCOLLEN BYTE)
                                (AROPCODE WORD)
                                (AR1STWORD WORD))
                (ACCESSFNS AR1STWORD ((ARCONTENTS (LOCF DATUM]

[ACCESSFNS ARETHER ((ARETHERBASE (fetch (AR ARCONTENTS) of DATUM)))
        (BLOCKRECORD ARETHERBASE ((ARLCLHDW0 WORD)
                                (ARLCLHDW1 WORD)
                                (ARLCLHDW2 WORD)
                                (ARLCLPTCL FIXP)
                                (ARFRNHDW0 WORD)
                                (ARFRNHDW1 WORD)
                                (ARFRNHDW2 WORD)
                                (ARFRNPTCL FIXP))
                [ACCESSFNS ARLCLHDW0 ((ARSENDERHDW (\LOADNSHOSTNUMBER (LOCF DATUM))
                                        (\STORENSHOSTNUMBER (LOCF DATUM)
                                                NEWVALUE]
                (ACCESSFNS ARFRNHDW0 ((ARTARGETHDW (\LOADNSHOSTNUMBER (LOCF DATUM))
                                        (\STORENSHOSTNUMBER (LOCF DATUM)
                                                NEWVALUE]

[ACCESSFNS AREXPETHER ((ARETHERBASE (fetch (AR ARCONTENTS) of DATUM)))
        (BLOCKRECORD ARETHERBASE ((ARLCLHDW WORD)
                                (ARLCLPTCL FIXP)
                                (ARFRNHDW WORD)
                                (ARFRNPTCL FIXP]

(DATATYPE ARENTRY ((RECENT FLAG)
                (SEARCHING FLAG)
                (IPADDRESS POINTER)
                (ETHERADDRESS POINTER)
                (TIMER POINTER))
        TIMER _ (NCREATE 'FIXP))
)
```

```
(/DECLAREDATATYPE 'ARENTRY '(FLAG FLAG POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '6)
```

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

(RPAQQ **\AR.HARDWARE.SPACE.ETHERNET** 1)

(RPAQQ **\AR.ETHERNET.ADDRESS.LENGTH** 6)

(RPAQQ **\AR.IP.ADDRESS.LENGTH** 4)

(RPAQQ **\AR.REQUEST** 1)

(RPAQQ **\AR.RESPONSE** 2)

(RPAQQ **\AR.ETHER.PACKET.LENGTH** 28)

```
(CONSTANTS (\AR.HARDWARE.SPACE.ETHERNET 1)
        (\AR.ETHERNET.ADDRESS.LENGTH 6)
        (\AR.IP.ADDRESS.LENGTH 4)
        (\AR.REQUEST 1)
        (\AR.RESPONSE 2)
        (\AR.ETHER.PACKET.LENGTH 28))
)
)
```

;; END EXPORTED DEFINITIONS

```
(/DECLAREDATATYPE 'ARENTRY '(FLAG FLAG POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '6)
```

(RPAQ? **\AR.IP.TO.10MB.ALIST** (CONS))

(RPAQ? **\AR.SEARCH.TIMEOUT.INTERVAL** 300000)

(RPAQ? **\AR.VALID.TIMEOUT.INTERVAL** 600000)

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

```
(GLOBALVARS \AR.IP.TO.10MB.ALIST \AR.SEARCH.TIMEOUT.INTERVAL \AR.VALID.TIMEOUT.INTERVAL)
)
```

```
(DEFINEQ
```

(**AR.DAEMON**
```
  [LAMBDA NIL                                                     (* ejs%: "25-Jun-85 18:47")
    (for ARENTRY in \AR.IP.TO.10MB.ALIST do (\AR.UPDATE.RESOLUTION ARENTRY)
                                        (BLOCK])
```

(**AR.ENTER.RESOLUTION**
```
  [LAMBDA (IPADDRESS ETHERADDRESS ONLY-IF-PRESENT-P)              ; Edited 21-Dec-88 20:10 by Briggs

            (* * Enter a new resolution in the AR table, or update an existing resolution)

    (LET [(OLDENTRY (find ENTRY in \AR.IP.TO.10MB.ALIST suchthat (EQUAL IPADDRESS (fetch (ARENTRY IPADDRESS)
                                                                                     of ENTRY]
          (COND
            (OLDENTRY (freplace (ARENTRY TIMER) of OLDENTRY with (SETUPTIMER \AR.VALID.TIMEOUT.INTERVAL
                                                                    (ffetch (ARENTRY TIMER) of OLDENTRY)))
                    (freplace (ARENTRY ETHERADDRESS) of OLDENTRY with ETHERADDRESS)
                    (freplace (ARENTRY RECENT) of OLDENTRY with T)
                    (freplace (ARENTRY SEARCHING) of OLDENTRY with NIL)
                    OLDENTRY)
            ((NOT ONLY-IF-PRESENT-P)
             (CAR (push \AR.IP.TO.10MB.ALIST (create ARENTRY
                                                    IPADDRESS _ IPADDRESS
                                                    ETHERADDRESS _ ETHERADDRESS
                                                    TIMER _ (SETUPTIMER \AR.VALID.TIMEOUT.INTERVAL)
                                                    RECENT _ T])
```

(**AR.NOTE.RESOLUTION**
```
  [LAMBDA (AR)                                                    ; Edited 21-Dec-88 20:11 by Briggs
```

;;; Use the information in the AR to update any existing entry in the cache, and if this was a response (presumably to our query) add the new information.

```
    [COND
        ((NOT (AND (EQ (fetch (ARETHER ARLCLHDW0) of AR)
                       0)
                   (EQ (fetch (ARETHER ARLCLHDW1) of AR)
                       0)
```

```
                            (EQ (fetch (ARETHER ARLCLHDW2) of AR)
                                0)))
            (\AR.ENTER.RESOLUTION (fetch (ARETHER ARLCLPTCL) of AR)
                  (fetch (ARETHER ARSENDERHDW) of AR)
                  (NOT (MEMBER (fetch (ARETHER ARFRNPTCL) of AR)
                            \IP.LOCAL.ADDRESSES]
        (COND
           ([AND (EQ (fetch (AR AROPCODE) of AR)
                     \AR.RESPONSE)
                 (NOT (AND (EQ (fetch (ARETHER ARFRNHDW0) of AR)
                               0)
                           (EQ (fetch (ARETHER ARFRNHDW1) of AR)
                               0)
                           (EQ (fetch (ARETHER ARFRNHDW2) of AR)
                               0]
            (\AR.ENTER.RESOLUTION (fetch (ARETHER ARFRNPTCL) of AR)
                  (fetch (ARETHER ARTARGETHDW) of AR])
```

(**AR.UPDATE.RESOLUTION**
```
  [LAMBDA (ARENTRY)                                                          ; Edited 21-Dec-88 18:27 by Briggs
```

;;; Called when a resolution is no longer recent. Does ARP requests to update our cache. Eventually, the entry is marked invalid and is removed

```
        (COND
           [(TIMEREXPIRED? (fetch (ARENTRY TIMER) of ARENTRY))
            (COND
               ((ffetch (ARENTRY RECENT) of ARENTRY)
                (freplace (ARENTRY RECENT) of ARENTRY with NIL)
                (freplace (ARENTRY SEARCHING) of ARENTRY with T)
                (freplace (ARENTRY TIMER) of ARENTRY with (SETUPTIMER \AR.SEARCH.TIMEOUT.INTERVAL (ffetch (ARENTRY TIMER
                                                                                                      )
                                                                                                  of ARENTRY)))
```

                ;; ask the system in the table to respond to avoid clogging the net with broadcasts

```
                (\AR.REQUEST.IP.TO.10MB (ffetch (ARENTRY IPADDRESS) of ARENTRY)
                      (ffetch (ARENTRY ETHERADDRESS) of ARENTRY)))
               ((ffetch (ARENTRY SEARCHING) of ARENTRY)
                (SETQ \AR.IP.TO.10MB.ALIST (DREMOVE ARENTRY \AR.IP.TO.10MB.ALIST]
           ((ffetch (ARENTRY SEARCHING) of ARENTRY)
            (\AR.REQUEST.IP.TO.10MB (ffetch (ARENTRY IPADDRESS) of ARENTRY)
                  (ffetch (ARENTRY ETHERADDRESS) of ARENTRY])
```

(**PRINTAR**
```
  [LAMBDA (AR CALLER FILE)                                             (* ejs%: " 2-Jun-85 13:58")
    (PROG NIL
          (SELECTC (fetch (ETHERPACKET EPTYPE) of AR)
                (\EPT.AR NIL)
                (3 (RETURN))
                (RETURN))
          (COND
             ((AND (EQ (fetch (AR ARHARDWARESPACE) of AR)
                       \AR.HARDWARE.SPACE.ETHERNET)
                   (EQ (fetch (AR ARHARDWARELEN) of AR)
                       \AR.ETHERNET.ADDRESS.LENGTH)
                   (EQ (fetch (AR ARPROTOCOLSPACE) of AR)
                       \EPT.IP)
                   (EQ (fetch (AR ARPROTOCOLLEN) of AR)
                       \AR.IP.ADDRESS.LENGTH))
              (printout FILE CALLER ": Address resolution " (SELECTC (fetch (AR AROPCODE) of AR)
                                                                 (\AR.REQUEST "request.")
                                                                 (\AR.RESPONSE "response.")
                                                                 "unknown opcode.")
                    T "Sender's protocol address is " (\IP.ADDRESS.TO.STRING (fetch (ARETHER ARLCLPTCL)
                                                                                 of AR))
                    "." T "Sender's hardware address is " (fetch (ARETHER ARSENDERHDW) of AR)
                    "." T)
              (SELECTC (fetch (AR AROPCODE) of AR)
                    (\AR.REQUEST (printout FILE "Sender desires hardware address for " (\IP.ADDRESS.TO.STRING
                                                                                         (fetch (ARETHER ARFRNPTCL)
                                                                                            of AR))
                                    T))
                    (\AR.RESPONSE (printout FILE "Sender says hardware address for " (\IP.ADDRESS.TO.STRING
                                                                                       (fetch (ARETHER ARFRNPTCL)
                                                                                          of AR))
                                     T "  is " (fetch (ARETHER ARTARGETHDW) of AR)
                                     T))
                    NIL)))
          (TERPRI FILE])
```

(**SPUTASSOC**
```
  [LAMBDA (KEY VAL ALIST)                                              (* ejs%: "27-Dec-84 17:52")
    (PROG (OLDENTRY)
          [COND
             ([SETQ OLDENTRY (for ENTRY in ALIST thereis (EQUAL KEY (CAR ENTRY]
```

```
            (RPLACD OLDENTRY VAL))
          (T (NCONC1 ALIST (CONS KEY VAL]
      (RETURN VAL))
```

(**AR.TRANSLATE.TO.10MB**
  [LAMBDA (IPADDRESS DONTPROBE)   ; Edited 21-Dec-88 20:11 by Briggs

;;; Translate an IPADDRESS to a 10MBHOSTNUMBER, or initiate request and fail for now

```
     (COND
        ((\IP.BROADCAST.ADDRESS IPADDRESS)
         BROADCASTNSHOSTNUMBER)
        [(bind FOUNDIT find ENTRY in \AR.IP.TO.10MB.ALIST suchthat (AND (EQUAL IPADDRESS (fetch (ARENTRY IPADDRESS)
                                                                                                of ENTRY))
                                                                        (SETQ FOUNDIT T))
              finally (COND
                        (FOUNDIT (RETURN (ffetch (ARENTRY ETHERADDRESS) of ENTRY]
        ((NOT DONTPROBE)
         (\AR.REQUEST.IP.TO.10MB IPADDRESS)
         NIL]))
```

(**AR.REQUEST.IP.TO.10MB**
  [LAMBDA (IPADDRESS PDH)   ; Edited 21-Dec-88 18:31 by Briggs

;;; Request an address translation, either from the specified host, or by broadcasting the request.

```
     (PROG ((AR (\ALLOCATE.ETHERPACKET)))
           (replace (AR ARHARDWARESPACE) of AR with \AR.HARDWARE.SPACE.ETHERNET)
           (replace (AR ARPROTOCOLSPACE) of AR with \EPT.IP)
           (replace (AR ARHARDWARELEN) of AR with \AR.ETHERNET.ADDRESS.LENGTH)
           (replace (AR ARPROTOCOLLEN) of AR with \AR.IP.ADDRESS.LENGTH)
           (replace (AR AROPCODE) of AR with \AR.REQUEST)
           (replace (ARETHER ARSENDERHDW) of AR with \MY.NSHOSTNUMBER)
           (replace (ARETHER ARLCLPTCL) of AR with (ffetch (NDB NDBIPHOST#) of \10MBLOCALNDB))
           (replace (ARETHER ARFRNPTCL) of AR with IPADDRESS)
           (replace (ETHERPACKET EPTYPE) of AR with \EPT.AR)
           (ENCAPSULATE.ETHERPACKET \10MBLOCALNDB AR (OR PDH BROADCASTNSHOSTNUMBER)
                  \AR.ETHER.PACKET.LENGTH \EPT.AR)
           (COND
              ((EQ IPTRACEFLG T)
               (PRINTPACKET AR 'PUT IPTRACEFILE))
              (IPTRACEFLG (PRIN1 (COND
                                   (PDH "!")
                                   (T "^"))
                                 IPTRACEFILE)))
           (TRANSMIT.ETHERPACKET \10MBLOCALNDB AR])
```

(**AR.REQUEST.IP.TO.3MB**
  [LAMBDA (IPADDRESS)   (* ejs%: " 2-Jan-85 17:12")

                    (* * Broadcast a request for an address translation)

```
     (PROG ((AR (\ALLOCATE.ETHERPACKET)))
           (replace (AR ARHARDWARESPACE) of AR with \AR.HARDWARE.SPACE.ETHERNET)
           (replace (AR ARPROTOCOLSPACE) of AR with \EET.IP)
           (replace (AR ARHARDWARELEN) of AR with 2)
           (replace (AR ARPROTOCOLLEN) of AR with \AR.IP.ADDRESS.LENGTH)
           (replace (AR AROPCODE) of AR with \AR.REQUEST)
           (replace (AREXPETHER ARLCLHDW) of AR with (LOGAND \LOCALPUPNETHOST (MASK.1'S 0 8)))
           (replace (AREXPETHER ARLCLPTCL) of AR with (ffetch (NDB NDBIPHOST#) of \3MBLOCALNDB))
           (replace (AREXPETHER ARFRNPTCL) of AR with IPADDRESS)
           (ENCAPSULATE.ETHERPACKET \3MBLOCALNDB AR 0 20 \EPT.AR)
           (COND
              (IPTRACEFLG (PRINTPACKET AR 'PUT IPTRACEFILE)))
           (TRANSMIT.ETHERPACKET \3MBLOCALNDB AR])
```

(**AR.RESOLVE**
  [LAMBDA (AR)   ; Edited 6-Jan-89 14:50 by Briggs

;;; Try to respond to an address resolution request. Release the packet if we can't

```
     (DECLARE (GLOBALVARS \10MBLOCALNDB \MY.NSHOSTNUMBER))
     (LET* ((TargetProtocolAddress (fetch (ARETHER ARFRNPTCL) of AR))
            (TargetHardwareAddress (COND
                                     ((MEMBER TargetProtocolAddress \IP.LOCAL.ADDRESSES)
                                      (\AR.ENTER.RESOLUTION TargetProtocolAddress \MY.NSHOSTNUMBER)
                                      \MY.NSHOSTNUMBER)
                                     ([AND \IP.GATEWAY.FLG
                                           (LET* ((SUBNETMASK (CDR (SASSOC (fetch NDBIPHOST# of \10MBLOCALNDB)
                                                                           \IP.SUBNET.MASKS)))
                                                  (MASKEDTARGET (LOGAND TargetProtocolAddress SUBNETMASK)))
                                                 (COND
                                                    ([AND SUBNETMASK (NOT (EQP MASKEDTARGET
```

```
                                                              (LOGAND (fetch NDBIPHOST#
                                                                        of \10MBLOCALNDB)
                                                                   SUBNETMASK]
                                         (for ADDRPAIR in \IP.ROUTING.TABLE when (LISTP ADDRPAIR)
                                           thereis (EQP MASKEDTARGET (CAR ADDRPAIR]
                                   \MY.NSHOSTNUMBER)))
               (SenderHardwareAddress (fetch (ARETHER ARSENDERHDW) of AR)))
           (COND
              (TargetHardwareAddress (swap (fetch (ARETHER ARLCLPTCL) of AR)
                                           (fetch (ARETHER ARFRNPTCL) of AR))
                  (replace (ARETHER ARTARGETHDW) of AR with (fetch (ARETHER ARSENDERHDW) of AR))
                  (replace (ARETHER ARSENDERHDW) of AR with TargetHardwareAddress)
                  (replace (ARETHER ARLCLPTCL) of AR with TargetProtocolAddress)
                  (replace (AR AROPCODE) of AR with \AR.RESPONSE)
                  (ENCAPSULATE.ETHERPACKET \10MBLOCALNDB AR SenderHardwareAddress \AR.ETHER.PACKET.LENGTH
                          \EPT.AR)
                  [COND
                     (IPTRACEFLG (COND
                                   ((EQ IPTRACEFLG T)
                                    (PRINTPACKET AR 'PUT IPTRACEFILE))
                                   (T (PRIN1 "!" IPTRACEFILE]
                  (TRANSMIT.ETHERPACKET \10MBLOCALNDB AR))
              (T (\RELEASE.ETHERPACKET AR])
```

AR.TRANSLATE.TO.3MB
```
  [LAMBDA (IPADDRESS)                                          (* ejs%: "27-Jun-85 12:43")
    (COND
       ((\IP.BROADCAST.ADDRESS IPADDRESS)
        0)
       (T (LDB (BYTE 8 0)
               IPADDRESS])
```

HANDLE.RAW.AR
```
  [LAMBDA (AR TYPE)                                            (* ejs%: " 2-Jun-85 14:12")
    (PROG ((NDB (ffetch (ETHERPACKET EPNETWORK) of AR)))
          (SELECTQ (ffetch (NDB NETTYPE) of NDB)
              (10 (COND
                     ((NEQ TYPE \EPT.AR)
                      (RETURN))))
              (3 (RETURN))
              (ERROR "Unknown net type" (fetch (NDB NETTYPE) of NDB)))
          [COND
             ((AND (EQ (fetch (AR ARHARDWARESPACE) of AR)
                       \AR.HARDWARE.SPACE.ETHERNET)
                   (EQ (fetch (AR ARHARDWARELEN) of AR)
                       \AR.ETHERNET.ADDRESS.LENGTH)
                   (EQ (fetch (AR ARPROTOCOLSPACE) of AR)
                       \EPT.IP)
                   (EQ (fetch (AR ARPROTOCOLLEN) of AR)
                       \AR.IP.ADDRESS.LENGTH))
              [COND
                 (IPTRACEFLG (COND
                               ((EQ IPTRACEFLG T)
                                (PRINTPACKET AR 'ARGET IPTRACEFILE))
                               (T (PRIN1 "*" IPTRACEFILE]
              (\AR.NOTE.RESOLUTION AR)
              (COND
                 ((EQ (fetch (AR AROPCODE) of AR)
                      \AR.REQUEST)
                  (\AR.RESOLVE AR))
                 (T (\RELEASE.ETHERPACKET AR]
          (RETURN T])
)

(ADDTOVAR \PACKET.PRINTERS (2054 . \PRINTAR))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX