```
(RPAQQ COURIERSERVECOMS
       ((DECLARE%: EVAL@COMPILE DONTCOPY (FILES UTILISOPRS ETHERRECORDS (LOADCOMP)
                                                COURIER))
        (FNS COURIER.START.SERVER COURIER.RETURN COURIER.ABORT COURIER.REJECT CLOSE.BULK.DATA
             COURIER.RESET.SOCKET)
        (FNS \COURIERSERVER \COURIER.SERVE \COURIER.APPLY \COURIER.SETUP.REPLY \COURIER.FINISH.REPLY
             \FIND.COURIER.PROGRAM \FIND.COURIER.PROCEDURE)
        (FNS \EXPEDITEDCOURIERSERVER \EXPEDITEDCOURIER.SERVE \CREATE.EXPEDITED.STREAM)
        (PROP ARGNAMES COURIER.CALL COURIER.EXPEDITED.CALL)
        (CONSTANTS (\REJECT.NOSUCHPROGRAM 0)
               (\REJECT.NOSUCHVERSION 1)
               (\REJECT.NOSUCHPROCEDURE 2)
               (\REJECT.INVALIDARGUMENTS 3)
               (\REJECT.UNSPECIFIED 65535)
               (\SINK.NULL 0)
               (\SINK.IMMEDIATE 1)
               (\SINK.PASSIVE 2)
               (\SINK.ACTIVE 3))
        (DECLARE%: DONTCOPY (RECORDS COURIERSERVERFN XIPSOCKETPAIR))))

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD UTILISOPRS ETHERRECORDS (LOADCOMP)
      COURIER)
)

(DEFINEQ
```

(**COURIER.START.SERVER**
```
  [LAMBDA (RESTART)                                                    ; Edited 21-Jul-88 14:44 by cdl
    (LET (PROCESS)
         (while (AND (SETQ PROCESS (FIND.PROCESS 'COURIER.LISTENER))
                     RESTART)
            do (DEL.PROCESS PROCESS)
               (BLOCK))
         (if (NULL PROCESS)
             then (COURIER.RESET.SOCKET)
                  (SPP.OPEN NIL \NS.WKS.Courier NIL 'COURIER.LISTENER '(SERVER.FUNCTION \COURIERSERVER
                                                                              OTHERXIPHANDLER
                                                                              \EXPEDITEDCOURIERSERVER))
                  (if COURIERTRACEFLG
                      then (printout COURIERTRACEFILE T "Courier Server started"))
                  (until (SETQ PROCESS (FIND.PROCESS 'COURIER.LISTENER)) do (BLOCK)))
         PROCESS])
```

(**COURIER.RETURN**
```
  [LAMBDA (COURIERSTREAM PROGRAM PROCEDURE RESULTLST)                  (* cdl "26-Nov-85 17:10")
    (LET (RESULTS STREAM)
         (SETQ RESULTS (with COURIERFN (\GET.COURIER.DEFINITION PROGRAM PROCEDURE 'PROCEDURES (
                                                                              \GET.COURIERPROGRAM
                                                                              PROGRAM))
                             RESULTS))
         (if (EQLENGTH RESULTLST (LENGTH RESULTS))
             then (SETQ STREAM (\COURIER.SETUP.REPLY COURIERSTREAM \COURIERMSG.RETURN))
                  (if (\COURIER.EXPEDITED.ARGS STREAM PROGRAM RESULTLST RESULTS)
                      then (\COURIER.FINISH.REPLY COURIERSTREAM STREAM)
                      else (COURIER.REJECT COURIERSTREAM))
             else (COURIER.REJECT COURIERSTREAM \REJECT.INVALIDARGUMENTS])
```

(**COURIER.ABORT**
```
  [LAMBDA (COURIERSTREAM PROGRAM ERROR RESULTLST)                      (* cdl "26-Nov-85 17:20")
    (LET [(STREAM (\COURIER.SETUP.REPLY COURIERSTREAM \COURIERMSG.ABORT))
          (ERRORFORM (CDR (ASSOC ERROR (fetch (COURIERPGM ERRORS) of (\GET.COURIERPROGRAM PROGRAM]
         (PUTWORD STREAM (CAR ERRORFORM))
         (if (\COURIER.EXPEDITED.ARGS STREAM PROGRAM RESULTLST (CADR ERRORFORM))
             then (\COURIER.FINISH.REPLY COURIERSTREAM STREAM)
             else (COURIER.REJECT \REJECT.INVALIDARGUMENTS])
```

(**COURIER.REJECT**
```
  [LAMBDA (COURIERSTREAM ERROR RESULTLST)                          (* cdl "26-Jun-87 10:13")
                                                                   (* DECLARATIONS%: (RECORD VERSIONRANGE
                                                                   (LOWEST . HIGHEST)))
      (LET ((STREAM (\COURIER.SETUP.REPLY COURIERSTREAM \COURIERMSG.REJECT)))
           [if (NOT (NUMBERP ERROR))
               then (SETQ ERROR (SELECTQ ERROR
                                    ((NIL UNSPECIFIED)
                                         \REJECT.UNSPECIFIED)
                                    (NO.SUCH.PROGRAM
                                         \REJECT.NOSUCHPROGRAM)
                                    (NO.SUCH.VERSION
                                         \REJECT.NOSUCHVERSION)
                                    (NO.SUCH.PROCEDURE
                                         \REJECT.NOSUCHPROCEDURE)
                                    (INVALID.ARGUMENTS
                                         \REJECT.INVALIDARGUMENTS)
                                    (SHOULDNT]
           (PUTWORD STREAM ERROR)
           (SELECTC ERROR
                (\REJECT.NOSUCHVERSION
                     (with VERSIONRANGE RESULTLST (PUTWORD STREAM LOWEST)
                           (PUTWORD STREAM HIGHEST)))
                NIL)
           (\COURIER.FINISH.REPLY COURIERSTREAM STREAM])
```

(**CLOSE.BULK.DATA**
```
  [LAMBDA (STREAM ABORTFLG)                                        ; Edited 21-Jul-88 15:18 by cdl

          (* Close a Bulk Data stream after the transfer has taken place. If a result function was specified in COURIER.CALL, call it on
          the stream and the result or list of results.)

      (PROG ((CON (fetch (SPPSTREAM SPP.CONNECTION) of STREAM)))
           (with SPPCON CON (SETQ SPPATTENTIONFN NIL)
               (if (NULL SPPSUBSTREAM)
                   then

          (* This stream has already been closed. We don't want to try to read the Courier results twice)

                        (RETURN)))
           (if (WRITEABLE STREAM)
               then (if ABORTFLG
                        then (SPP.SENDATTENTION STREAM 1)
                      else (SPP.SENDEOM STREAM))
             elseif (NOT (\EOFP STREAM))
               then                                                (* Closing before all the data has been read --
                                                                   abort the transfer.)

                    (if (NOT ABORTFLG)
                        then (SETQ ABORTFLG T))
                    (\ABORT.BULK.DATA STREAM))
           (SPP.DSTYPE STREAM \SPPDSTYPE.COURIER)
           (with SPPCON CON (SETQ SPPINPKT NIL)
               (SETQ SPPSUBSTREAM NIL))

          (* This stream is closing; make sure there aren't any dangling pointers into the middle of ether packets.)

           (with STREAM STREAM (SETQ CBUFPTR NIL)
               (SETQ CBUFSIZE 0))

          (* The result of the Courier call may be an error which the user should see;
          however, we still need to clean up the substream, so we wrap it in this RESETLST.)

           (if ABORTFLG
               then (with STREAM (with SPPCON CON SPPINPUTSTREAM)
                        (SETQ ENDOFSTREAMOP (FUNCTION \COURIER.EOF)))
                    (with STREAM STREAM (SETQ ENDOFSTREAMOP (FUNCTION ERROR!])
```

(**COURIER.RESET.SOCKET**
```
  [LAMBDA NIL                                                      ; Edited  5-Apr-88 14:08 by cdl
      (CLOSENSOCKET (OPENNSOCKET \NS.WKS.Courier 'ACCEPT)
           T])
```

)

(DEFINEQ

(**COURIERSERVER**
```
  [LAMBDA (STREAM)                                                ; Edited  5-Apr-88 14:02 by cdl
      (DECLARE (SPECVARS STREAM))
      (RESETLST
          (RESETSAVE NIL '(CLOSEF? %, STREAM))
          [PROG (LOW.VERSION HIGH.VERSION)
                (SPP.DSTYPE STREAM \SPPDSTYPE.COURIER)
                (with STREAM STREAM (SETQ ENDOFSTREAMOP (FUNCTION \COURIER.EOF)))
                (PROCESS.NAME (THIS.PROCESS
```

```
                       (PACK* 'COURIER (SPP.DESTADDRESS STREAM)))
               (if COURIERTRACEFLG
                  then (printout COURIERTRACEFILE T "Server opened connection with" %, (SPP.DESTADDRESS STREAM))
                       )
               (PUTWORD STREAM COURIER.VERSION#)
               (PUTWORD STREAM COURIER.VERSION#)
               (SPP.FORCEOUTPUT STREAM)
               (SETQ LOW.VERSION (GETWORD STREAM))
               (SETQ HIGH.VERSION (GETWORD STREAM))
               (if (AND (LEQ LOW.VERSION COURIER.VERSION#)
                        (LEQ COURIER.VERSION# HIGH.VERSION))
                  then (SPP.CLEAREOM STREAM T)
                       (\COURIER.SERVE STREAM)
                else (if COURIERTRACEFLG
                         then (printout COURIERTRACEFILE T "Client requesting unknown version of Courier" %,
                                        (SPP.DESTADDRESS STREAM)
                                        %, LOW.VERSION %, HIGH.VERSION])])
```

(**COURIER.SERVE**

```
  [LAMBDA (STREAM)                                                    (* cdl "30-Jun-87 08:57")
    (until (SPP.EOFP STREAM)
       do
       (SELECTC (GETWORD STREAM)
             (\COURIERMSG.CALL
                 (PROG (PROGRAM PROGRAM# VERSION# PROCEDURE# PROCEDURE PGMDEF PROCDEF VERSIONS RESULTLST)
                       (STREAMPROP STREAM 'TRANSACTIONID (GETWORD STREAM))
                                                            (* Save the Transaction ID.)
                       (SETQ PROGRAM# (GETLONG STREAM))
                       (SETQ VERSION# (GETWORD STREAM))
                       (SETQ PROCEDURE# (GETWORD STREAM))
                       (if (SETQ VERSIONS (\FIND.COURIER.PROGRAM PROGRAM# VERSION#))
                          then [if (SETQ PROGRAM (CDR (FASSOC VERSION# VERSIONS)))
                                  then (if (SETQ PROCEDURE (\FIND.COURIER.PROCEDURE (SETQ PGMDEF (
                                                                                            \GET.COURIERPROGRAM
                                                                                            PROGRAM))
                                                          PROCEDURE#))
                                          then (SETQ PROCDEF (\GET.COURIER.DEFINITION PROGRAM PROCEDURE
                                                                   'PROCEDURES PGMDEF))
                                               (with COURIERSERVERFN PROCDEF
                                                     (if SERVERFN
                                                        then (if (NEQ (\COURIER.APPLY SERVERFN STREAM PROGRAM
                                                                              PROCEDURE PGMDEF PROCDEF)
                                                                      'FLUSH)
                                                                then (RETURN))
                                                      else (COURIER.REJECT STREAM \REJECT.NOSUCHPROCEDURE)))
                                        else (COURIER.REJECT STREAM \REJECT.NOSUCHPROCEDURE))
                                else (COURIER.REJECT STREAM \REJECT.NOSUCHVERSION (CONS (in VERSIONS minimize
                                                                                            CAR)
                                                                                       (in VERSIONS maximize
                                                                                            CAR]
                          else (COURIER.REJECT STREAM \REJECT.NOSUCHPROGRAM))
                                                            (* Flush arguments from rejected call)
                       (bind FLG until (SETQ FLG (SPP.EOFP STREAM)) do (BIN STREAM)
                          finally (SELECTQ FLG
                                      (EOM (SPP.CLEAREOM STREAM))
                                      NIL))))
             (SHOULDNT])
```

(**COURIER.APPLY**

```
  [LAMBDA (FN STREAM PROGRAM PROCEDURE PGMDEF PROCDEF)            (* cdl "26-Jun-87 11:37")
    (DECLARE (SPECVARS FN STREAM PROGRAM PROCEDURE PGMDEF PROCDEF))
    (RESETLST
        (PROG (XIPSOCKETPAIR ARGLST RESULTS DATASTREAM)
              (DECLARE (SPECVARS XIPSOCKETPAIR ARGLST))
             [if (type? XIPSOCKETPAIR STREAM)
                then (with XIPSOCKETPAIR (SETQ XIPSOCKETPAIR STREAM)
                           (SETQ STREAM (\CREATE.EXPEDITED.STREAM XIP 'INPUT NIL (INDEXF (fetch (EXPEDITEDXIP
                                                                                                  ARG0)
                                                                                          of T]
              [SETQ ARGLST (for TYPE in (with COURIERFN PROCDEF ARGS)
                              collect (SELECTQ TYPE
                                          ((BULK.DATA.SINK BULK.DATA.SOURCE)
                                           (SELECTC (GETWORD STREAM)
                                                 (\SINK.IMMEDIATE
                                                     NIL)
                                                 ((LIST \SINK.NULL \SINK.PASSIVE \SINK.ACTIVE)
                                                            (* Server only handles IMMEDIATE transfer, error out as cleanly
                                                              as we can)
                                                     (COURIER.REJECT STREAM)
                                                     (RETURN 'FLUSH))
                                                 (SHOULDNT))
                                           (SETQ DATASTREAM (\BULK.DATA.STREAM STREAM
                                                                   (SELECTQ TYPE
                                                                       (BULK.DATA.SOURCE
                                                                          'INPUT)
```

```
                                                                      'OUTPUT)
                                                       PROGRAM PROCEDURE PGMDEF PROCDEF)))
                                   (COURIER.READ STREAM PROGRAM TYPE]
                  (if XIPSOCKETPAIR
                      then (SETQ STREAM XIPSOCKETPAIR)
                    else (SELECTQ (SPP.EOFP STREAM)
                                 (EOM (SPP.CLEAREOM STREAM))
                                 NIL))
                  (if [SETQ RESULTS (NLSETQ (APPLY FN (LIST* STREAM PROGRAM PROCEDURE ARGLST]
                      then (SETQ RESULTS (CAR RESULTS))
                    else (if COURIERTRACEFLG
                             then (printout COURIERTRACEFILE T "Courier server function break:" %, FN))
                         (COURIER.REJECT STREAM \REJECT.UNSPECIFIED)
                         (RETURN))
                  (if (AND DATASTREAM (OPENP DATASTREAM))
                      then (CLOSE.BULK.DATA DATASTREAM))
                  (SELECTQ (CAR RESULTS)
                      (RETURN (COURIER.RETURN STREAM PROGRAM PROCEDURE (CDR RESULTS)))
                      (ABORT (COURIER.ABORT STREAM PROGRAM (CADR RESULTS)
                                  (CDDR RESULTS)))
                      (REJECT (COURIER.REJECT STREAM (CADR RESULTS)
                                  (CDDR RESULTS)))
                      NIL)
                  (RETURN)))])
```

(**COURIER.SETUP.REPLY**
```
  [LAMBDA (STREAM TYPE)                                     (* cdl "26-Jun-87 11:30")
    (LET (TRANSACTIONID)
        [if (type? XIPSOCKETPAIR STREAM)
            then (SETQ TRANSACTIONID (with EXPEDITEDXIP (with XIPSOCKETPAIR STREAM XIP)
                                            TRANSACTIONID))
                 [with XIPSOCKETPAIR STREAM (SETQ STREAM (\CREATE.EXPEDITED.STREAM XIP 'OUTPUT NIL
                                                            (INDEXF (fetch (EXPEDITEDXIP MSGTYPE)
                                                                      of T]
          else (SETQ TRANSACTIONID (STREAMPROP STREAM 'TRANSACTIONID]
        (PUTWORD STREAM TYPE)
        (PUTWORD STREAM TRANSACTIONID)                         (* Transaction ID just echoed for now.)
        STREAM])
```

(**COURIER.FINISH.REPLY**
```
  [LAMBDA (COURIERSTREAM STREAM)                            (* cdl " 7-Nov-85 19:22")
    (if (type? XIPSOCKETPAIR COURIERSTREAM)
        then (with XIPSOCKETPAIR COURIERSTREAM (replace (XIP XIPLENGTH) of XIP with (with STREAM STREAM COFFSET))
                   (SWAPXIPADDRESSES XIP)
                   (with ETHERPACKET XIP (SETQ EPREQUEUE 'FREE))
                   (SENDXIP SOCKET XIP))
      else (SPP.SENDEOM COURIERSTREAM])
```

(**FIND.COURIER.PROGRAM**
```
  [LAMBDA (PROGRAM VERSION)                                 (* cdl "30-Jun-87 08:56")
    (DECLARE (SPECVARS PROGRAM VERSION))
    (LET (VERSIONS)
        (DECLARE (SPECVARS VERSIONS))
        [MAPHASH \COURIERPROGRAM (FUNCTION (LAMBDA (VALUE KEY)
                                      (DECLARE (USEDFREE PROGRAM VERSION VERSIONS))
                                      (with COURIERPGM VALUE
                                          (if (EQP PROGRAM# PROGRAM)
                                              then (push VERSIONS (CONS VERSION# KEY))
                                                   (if (EQP VERSION# VERSION)
                                                       then
                                                       (* We found the requested version, no reason to look for others)
                                                            (RETFROM (FUNCTION MAPHASH]
        VERSIONS])
```

(**FIND.COURIER.PROCEDURE**
```
  [LAMBDA (PGMDEF PROCEDURE#)                               (* cdl "26-Jun-87 08:33")
    (LET (PROCEDURE)
        (if [SETQ PROCEDURE (OR (for PROCEDURE in (with COURIERPGM PGMDEF PROCEDURES)
                                   thereis (EQP (with COURIERFN (CDR PROCEDURE)
                                                      FN#)
                                              PROCEDURE#))
                                (for OTHERPROGRAM in (with COURIERPGM PGMDEF INHERITS)
                                   when [SETQ $$VAL (with COURIERPGM (\GET.COURIERPROGRAM OTHERPROGRAM)
                                                          (for PROCEDURE in PROCEDURES
                                                             thereis (EQP (with COURIERFN (CDR PROCEDURE)
                                                                                FN#)
                                                                        PROCEDURE#]
                                   do (RETURN $$VAL]
            then (CAR PROCEDURE]
```

```
)

(DEFINEQ
```

(**EXPEDITEDCOURIERSERVER**
```
  [LAMBDA (XIP SOCKET)                                            (* cdl "26-Jun-87 09:08")
    (if COURIERTRACEFLG
        then (printout COURIERTRACEFILE T "Expedited server connection with" %, (with XIP XIP XIPSOURCENSADDRESS)
                ))
    (LET ((XIPSOCKETPAIR (create XIPSOCKETPAIR
                                 XIP _ XIP
                                 SOCKET _ SOCKET)))
         (with EXPEDITEDXIP XIP (if (NOT (AND (LEQ LOWVERSION COURIER.VERSION#)
                                              (LEQ COURIER.VERSION# HIGHVERSION)))
                                    then (if COURIERTRACEFLG
                                             then (printout COURIERTRACEFILE T "Client requesting unknown version
                                                          of Courier" %, (with XIP XIP XIPSOURCENSADDRESS)
                                                          %, LOWVERSION %, HIGHVERSION))
                                         (COURIER.REJECT XIPSOCKETPAIR)))
         (ADD.PROCESS '(\EXPEDITEDCOURIER.SERVE %, (KWOTE XIPSOCKETPAIR]
```

(**EXPEDITEDCOURIER.SERVE**
```
  [LAMBDA (XIPSOCKETPAIR)                                          (* cdl "30-Jun-87 08:57")
    (with EXPEDITEDXIP (with XIPSOCKETPAIR XIPSOCKETPAIR XIP)
          (SELECTC MSGTYPE
               (\COURIERMSG.CALL
                    (PROG (PROGRAM PROCEDURE PGMDEF PROCDEF VERSIONS RESULTLST)
                          (if (SETQ VERSIONS (\FIND.COURIER.PROGRAM PROGRAM# VERSION#))
                              then [if (SETQ PROGRAM (CDR (FASSOC VERSION# VERSIONS)))
                                       then (if (SETQ PROCEDURE (\FIND.COURIER.PROCEDURE (SETQ PGMDEF (
                                                                                              \GET.COURIERPROGRAM
                                                                                                PROGRAM))
                                                                        PROCEDURE#))
                                                then (SETQ PROCDEF (\GET.COURIER.DEFINITION PROGRAM PROCEDURE
                                                                          'PROCEDURES PGMDEF))
                                                     (with COURIERSERVERFN PROCDEF
                                                           (if SERVERFN
                                                               then (\COURIER.APPLY SERVERFN XIPSOCKETPAIR PROGRAM
                                                                             PROCEDURE PGMDEF PROCDEF)
                                                               else (COURIER.REJECT XIPSOCKETPAIR
                                                                            \REJECT.NOSUCHPROCEDURE)))
                                                else (COURIER.REJECT XIPSOCKETPAIR \REJECT.NOSUCHPROCEDURE))
                                       else (COURIER.REJECT XIPSOCKETPAIR \REJECT.NOSUCHVERSION
                                                    (CONS (in VERSIONS minimize CAR)
                                                          (in VERSIONS maximize CAR]
                              else (COURIER.REJECT XIPSOCKETPAIR \REJECT.NOSUCHPROGRAM))))
               (SHOULDNT])
```

(**CREATE.EXPEDITED.STREAM**
```
  [LAMBDA (XIP ACCESS OSTREAM OFFSET)                              ; Edited 21-Jul-88 15:08 by cdl

            (* * Makes a STREAM to access the contents of XIP as an expedited courier message body.
            We use the BASEBYTES device for simplicity. All the operations we actually need are BIN, BOUT, BLOCKIN and
            BLOCKOUT)

    (LET ([STREAM (OR OSTREAM (NCREATE 'STREAM]
              END)
         (replace (STREAM DEVICE) of STREAM with \BASEBYTESDEVICE)
         (replace (STREAM ACCESS) of STREAM with ACCESS)
         (replace (STREAM CBUFPTR) of STREAM with (fetch (XIP XIPBASE) of XIP))
         [replace (STREAM COFFSET) of STREAM with (IPLUS \XIPOVLEN (UNFOLD 3 BYTESPERWORD)
                                                         (if (EQ ACCESS 'INPUT)
                                                             then  (* For COURIER.RESULTS)
                                                                   (SETQ END (fetch (XIP XIPLENGTH) of XIP))
                                                                   (UNFOLD (OR OFFSET (INDEXF (fetch (EXPEDITEDXIP
                                                                                                       MSGTYPE)
                                                                                               of T)))
                                                                           BYTESPERWORD)
                                                             else  (* For COURIER.EXPEDITED.ARGS)
                                                                   (SETQ END (IPLUS \MAX.XIPDATALENGTH \XIPOVLEN))
                                                                   (UNFOLD (OR OFFSET (INDEXF (fetch (EXPEDITEDXIP
                                                                                                       ARG0)
                                                                                               of T)))
                                                                           BYTESPERWORD]
         (replace (STREAM EOFFSET) of STREAM with (replace (STREAM CBUFSIZE) of STREAM with END))
         [if (EQ ACCESS 'INPUT)
             then

                  (* Will cause error if COURIER.RESULTS tries to read more than was sent --
                  should never happen)

                  (with STREAM STREAM (SETQ ENDOFSTREAMOP (FUNCTION \COURIER.EOF)))
             else                                                  (* Invoked if COURIER.EXPEDITED.ARGS tries to write more
                                                                   than will fit in the packet)
                  (with BASEBYTESTREAM STREAM (SETQ WRITEXTENSIONFN (FUNCTION \COURIER.EXPEDITED.OVERFLOW]
         STREAM])

)
```

(PUTPROPS **COURIER.CALL ARGNAMES** (NIL (STREAM PROGRAM PROCEDURE ARG1 |...| ARGn NOERRORFLG) . U))

(PUTPROPS **COURIER.EXPEDITED.CALL ARGNAMES** (NIL (ADDRESS SOCKET PROGRAM PROCEDURE ARG1 |...| ARGn NOERRORFLG)
)
                                                                                                      . U)

(DECLARE%: EVAL@COMPILE

(RPAQQ **\REJECT.NOSUCHPROGRAM** 0)

(RPAQQ **\REJECT.NOSUCHVERSION** 1)

(RPAQQ **\REJECT.NOSUCHPROCEDURE** 2)

(RPAQQ **\REJECT.INVALIDARGUMENTS** 3)

(RPAQQ **\REJECT.UNSPECIFIED** 65535)

(RPAQQ **\SINK.NULL** 0)

(RPAQQ **\SINK.IMMEDIATE** 1)

(RPAQQ **\SINK.PASSIVE** 2)

(RPAQQ **\SINK.ACTIVE** 3)

(CONSTANTS (\REJECT.NOSUCHPROGRAM 0)
        (\REJECT.NOSUCHVERSION 1)
        (\REJECT.NOSUCHPROCEDURE 2)
        (\REJECT.INVALIDARGUMENTS 3)
        (\REJECT.UNSPECIFIED 65535)
        (\SINK.NULL 0)
        (\SINK.IMMEDIATE 1)
        (\SINK.PASSIVE 2)
        (\SINK.ACTIVE 3))
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD COURIERSERVERFN (FN# ARGS RETURNSNOISE RESULTS REPORTSNOISE ERRORS IMPLEMENTEDNOISE SERVERFN))

(TYPERECORD XIPSOCKETPAIR (XIP . SOCKET))
)
)

(PUTPROPS **COURIERSERVE COPYRIGHT** ("Xerox Corporation & Stanford University" 1985 1986 1987 1988 1991))

FUNCTION INDEX

CONSTANT INDEX

RECORD INDEX

PROPERTY INDEX